

# A Unifying Framework for Devising Efficient and Irreversible MCMC Samplers

Yi-An Ma<sup>1</sup>, Emily B. Fox<sup>2</sup>, Tianqi Chen<sup>3</sup>, and Lei Wu<sup>4</sup>

<sup>1</sup>Department of Applied Mathematics

<sup>2</sup>Department of Statistics

<sup>3</sup>Department of Computer Science and Engineering, University of Washington

<sup>4</sup>School of Mathematical Sciences, Peking University

December 5, 2016

## Summary

We propose a framework for Markov chain Monte Carlo using both continuous dynamics and jump processes that enable the development of efficient, irreversible samplers. For each component, we decompose the dynamics into reversible and irreversible processes, and with a parameterization that is easy to specify while ensuring the correct stationary distribution. Furthermore, we devise easy-to-implement, practical algorithms. For the continuous dynamic processes, our sampler is akin to HMC, but handling a complete family of stochastic dynamics. For our jump processes, we propose an implementation akin to Metropolis-Hastings—with the same ease of implementation—but allowing for irreversible dynamics. In our experiments, we demonstrate the benefits of such irreversibility across a broad range of target distributions. For both our continuous and jump frameworks, we discuss scalable variants for large-scale Bayesian inference. Finally, we combine the frameworks and show how one can use the continuous dynamics as a proposal distribution in the irreversible jump sampler. This can be viewed either as a method for correcting for bias in the discretized continuous dynamics or as a method for incorporating geometric information into the irreversible jump sampler.

*Keywords:* Bayesian inference; Hamiltonian Monte Carlo; Irreversible samplers; Jump processes; Markov chain Monte Carlo; Metropolis-Hastings

## 1 Introduction

Markov chain Monte Carlo (MCMC) methods are the defacto tools for inference in Bayesian models [Liu, 2004, Robert and Casella, 2004]. There are two primary approaches to developing and implementing such MCMC algorithms. One is the traditional Metropolis-Hastings type of approach, where one defines a jump process through an accept-reject procedure. This popular class of methods utilizes global information of the target distribution. Another approach relies on the local (gradient) information of the target distribution and designs a continuous dynamical process with the target distribution as its stationary distribution; samples are proposed according to the integrated trajectories of the continuous dynamics. Important examples of such samplers include Hamiltonian Monte Carlo methods [Duane et al., 1987, Neal, 2010] and samplers using Langevin dynamics [Roberts and Stramer, 2002, Xifara et al., 2014, Welling and Teh, 2011, Patterson and Teh, 2013].

For both approaches, a major challenge is devising a sampler with good mixing rates (i.e., the speed at which an initial distribution converges to the target distribution). In the world of jump-process-based MCMC techniques,

a focus has been on developing clever proposals [Tak et al., 2016, Jarner and Roberts, 2007, Liu, 2004], but these methods are often strongly coupled to a specific challenge setting, like multimodal targets [Tak et al., 2016] or heavy tailed distributions [Jarner and Roberts, 2007]. In practice, one often does not know the structure of the target distribution, which might additionally exhibit a combination of these factors. In the world of continuous-dynamic-based samplers, methods using second order information, like Riemannian Hamiltonian Monte Carlo [Girolami and Calderhead, 2011], can be helpful. However, it is non-trivial to devise these modifications and prove that the dynamics maintain the right stationary distribution.

In this paper, we propose a unifying framework for these two approaches that enables a more user-friendly method for devising efficient and general-purpose MCMC procedures. We start by examining continuous dynamic samplers. We present a stochastic differential equation (SDE) based framework in which to define all such valid samplers based on specifying two matrices: a positive semidefinite diffusion matrix and a skew-symmetric curl matrix. These matrices define symmetric (reversible) and anti-symmetric (irreversible) operators, respectively. Based on this framework, we prove that for any choice of these matrices, the sampler will have the target distribution as the stationary distribution. We likewise prove that any continuous dynamic sampler with the correct stationary distribution has a representation in this framework. As such, we call this framework *complete*. We cast a number of past methods in our proposed representation, and also show how it can be used to devise new samplers with improved mixing rates. An initial version of this work appeared in [Ma et al., 2015].

In [Ma et al., 2015], jump processes were specifically excluded from the analysis. However, jump processes represent a potentially attractive approach to MCMC since, in theory, samples generated from jump processes can decorrelate rapidly. In practice, it is challenging to define transition kernels that enable this efficient exploration while maintaining a reasonably high acceptance rate. The challenge partially stems from the fact that attention has typically been restricted to *reversible* jump processes. Such samplers are straightforward to derive and implement, but the reversibility restriction hinders the mixing rates and efficiency of the proposed algorithms [Neal, 2004, Diaconis et al., 2000, Chen and Hwang, 2013, Chen et al., 1999]. Unfortunately, devising irreversible samplers is a non-trivial task, and often results in computationally complex algorithms.

Leveraging our insights from the operator decomposition for continuous dynamic processes, we show that a similar framework can enable the development of efficient *irreversible* jump process samplers based on the specification of two kernel functions. We decompose the jump operator into symmetric (reversible) and anti-symmetric (irreversible) operators, paralleling the continuous-dynamic sampler framework. Using this decomposition and parameterization, we arrive at a straightforward set of constraints on the transition kernels that ensures that the target distribution is the stationary distribution. The resulting sampler implementation has the ease and efficiency of Metropolis-Hastings; in fact, the implementation directly parallels that of standard Metropolis-Hastings. In terms of runtime, our proposed method significantly outperforms previous approaches [Metropolis et al., 1953, Hastings, 1970], in addition to providing fast mixing rates in a range of scenarios, from heavy-tailed to multimodal targets. We demonstrate these performance gains against existing approaches in a variety of sampling tasks.

There are many ways we can think of combining our continuous dynamic and jump process frameworks. One is to use the continuous dynamic sampler and jump process sampler iteratively, i.e., use one (possibly for multiple iterations) and then the other, just as in Hamiltonian Monte Carlo. Another approach is to use the continuous dynamic sampler for some variables (e.g., real-valued variables) and the jump process sampler for others (e.g., discrete-valued variables). It is straightforward to combine our approaches in these manners since each process maintains the correct stationary distribution, so these types of compositions will likewise result in the correct stationary distribution under some mild conditions at stationarity. The strategy of alternating between continuous dynamics and jump processes is similar to what was recently proposed in the *bouncy particle* [Bouchard-Côté et al., 2016] and *Zig-Zag* [Bierkens and Roberts, 2016, Bierkens et al., 2016] samplers. These samplers iterate between deterministic continuous dynamics and Poisson jump processes. However, the algorithms associated with these methods are quite involved and deviate significantly from classical MCMC tools.

Alternatively, as we show, we are able to use a discretization of the continuous dynamics as a proposal distribution in our jump process accept-reject scheme, even when the continuous dynamics are not reversible. Here,

the transition kernel is defined according to the SDE representation of the continuous dynamics. Importantly, the simplicity of the Metropolis-Hastings algorithm is inherited. We can view the benefits of this approach from two angles: (i) the SDE can provide an efficient proposal distribution for our irreversible Markov jump process or (ii) the accept-reject scheme allows us to correct for the bias introduced by sampling the continuous dynamics via a discretized SDE. This accept-reject scheme is a direct generalization of the Metropolis-adjusted Langevin algorithm (MALA) [Roberts and Stramer, 2002] to irreversible SDEs. This opens up the possibility to combine, for example, the benefits of Langevin diffusion with Hamiltonian dynamics. As we see, combining our frameworks for continuous dynamic processes and Markov jump processes yields a unified and complete framework for devising efficient and general purpose MCMC samplers with the correct stationary distribution.

We conclude with a discussion on how to scale our continuous and jump frameworks to perform Bayesian inference in large datasets.

## 2 Background

We start with the standard MCMC goal of drawing samples from a target distribution  $\pi(\mathbf{z})$ . The idea behind MCMC is to translate the task of sampling from the posterior distribution to simulating from a Markov process. One can then discuss the evolution of the distribution on  $\mathbf{z}$  at time  $t$ ,  $p(\mathbf{z}; t)$ , under this stochastic process and analyze its *stationary distribution*,  $p^s(\mathbf{z})$ . If the stochastic process is ergodic and its stationary distribution is equal to the target distribution  $\pi(\mathbf{z})$ , then simulating the stationary stochastic dynamics equates with providing samples from the posterior distribution.

In this section, we review some of the fundamentals of the stochastic processes associated with general Markov processes, and how we can use these processes to construct samplers.

### 2.1 General Markov Processes and their Diffusion and Jump Operators

Consider a general Markov process in  $\mathbb{R}^d$ , described by the Chapman-Kolmogorov (CK) equation:

$$p(\mathbf{z}|\mathbf{x}; t_1 + t_2) = \int_{\mathbb{R}^d} d\mathbf{y} p(\mathbf{z}|\mathbf{y}; t_2) p(\mathbf{y}|\mathbf{x}; t_1), \quad (1)$$

where  $t_1 < t_2$  are two arbitrary scalar variables denoting time and  $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{R}^d$ . For succinctness of presentation, we let  $p(\mathbf{y}|\mathbf{x}; t)$  denote the probability of transition from  $\mathbf{x}$  to  $\mathbf{y}$  over time  $t$  and assume autonomous Markov processes (i.e., with time invariant Markov transition operators). It is worth noting that the autonomous assumption is not necessary to any of the calculations; non-autonomous Markov processes bear exactly the same results.

A differential form of (1), from which algorithms are more straightforwardly derived, can be obtained by assuming three mild existence conditions specified in Appendix A.1. The differential CK equation defines an update rule that consists of a diffusion process and a jump process [Gardiner, 2009]:

$$\begin{aligned} \frac{\partial}{\partial t} p(\mathbf{z}|\mathbf{y}; t) &= \sum_{i,j=1}^d \frac{\partial^2}{\partial \mathbf{z}_i \partial \mathbf{z}_j} [D_{ij}(\mathbf{z}) p(\mathbf{z}|\mathbf{y}; t)] - \sum_{i=1}^d \frac{\partial}{\partial \mathbf{z}_i} [\mathbf{f}_i(\mathbf{z}) p(\mathbf{z}|\mathbf{y}; t)] \\ &\quad + \int_{\mathbb{R}^d} d\mathbf{x} [W(\mathbf{z}|\mathbf{x}) p(\mathbf{x}|\mathbf{y}; t) - W(\mathbf{x}|\mathbf{z}) p(\mathbf{z}|\mathbf{y}; t)]. \end{aligned} \quad (2)$$

Here,  $\mathbf{f} : \mathbb{R}^d \rightarrow \mathbb{R}^d$ ,  $W : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$  is a transition probability rate function (which defines the transition kernel in (8)), and  $D$  is a  $d \times d$  positive semidefinite matrix. The first line denotes a continuous Markov process specified by a Fokker-Planck operator on  $p(\mathbf{z}|\mathbf{y}; t)$ ; the second line is a jump process defined by the transition rate function  $W(\mathbf{z}|\mathbf{x})$ .

In this paper, we rewrite (2) as

$$\frac{\partial}{\partial t} p(\mathbf{z}|\mathbf{y}; t) = \hat{\mathcal{L}} \left[ \frac{p(\mathbf{z}|\mathbf{y}; t)}{\pi(\mathbf{z})} \right] + \hat{\mathcal{J}} \left[ \frac{p(\mathbf{z}|\mathbf{y}; t)}{\pi(\mathbf{z})} \right], \quad (3)$$

where  $\hat{\mathcal{L}}[\cdot]$  is a diffusion operator defined as:

$$\hat{\mathcal{L}}[\varphi(\mathbf{z})] = \sum_{i,j=1}^d \frac{\partial^2}{\partial \mathbf{z}_i \partial \mathbf{z}_j} [D_{ij}(\mathbf{z}) \pi(\mathbf{z}) \varphi(\mathbf{z})] - \sum_{i=1}^n \frac{\partial}{\partial \mathbf{z}_i} [\mathbf{f}_i(\mathbf{z}) \pi(\mathbf{z}) \varphi(\mathbf{z})], \quad (4)$$

and  $\hat{\mathcal{J}}[\cdot]$  is a Markov transition (jump) operator with kernel  $W(\mathbf{z}|\mathbf{x})$ :

$$\hat{\mathcal{J}}[\varphi(\mathbf{z})] = \int_{\mathbb{R}^d} d\mathbf{x} [W(\mathbf{z}|\mathbf{x}) \pi(\mathbf{x}) \varphi(\mathbf{x}) - W(\mathbf{x}|\mathbf{z}) \pi(\mathbf{z}) \varphi(\mathbf{z})]. \quad (5)$$

This form allows us to do two things straightforwardly: One is to separate our analyses of the continuous and jump parts; the second is to analyze the reversibility of the processes. In particular, reversible processes satisfy the algebraic condition that  $\frac{p(\mathbf{x}|\mathbf{y}; t)}{\pi(\mathbf{x})} = \frac{p(\mathbf{y}|\mathbf{x}; t)}{\pi(\mathbf{y})}$ , or as is more commonly written,  $p(\mathbf{x}|\mathbf{y}; t) \pi(\mathbf{y}) = p(\mathbf{y}|\mathbf{x}; t) \pi(\mathbf{x})$ .

When the operators  $\hat{\mathcal{L}}$  and  $\hat{\mathcal{J}}$  on  $p(\mathbf{z}|\mathbf{y}; t)/\pi(\mathbf{z})$  are *self-adjoint* (i.e., their adjoint operators in the Hilbert space  $L^2(\mathbb{R})$  are equal to themselves), the Markov process is reversible. Hence, expressing the evolution of the current probability distribution with respect to the stationary distribution as in (3) can help reveal this structure.

For simplicity, to ensure the correctness of the resulting samplers, in this paper we assume that individually the continuous and jump Markov processes each have  $\pi(\mathbf{z})$  as their invariant solution (i.e.,  $\pi(\mathbf{z})$  is in the null space of both operators). Thus, when combined,  $\pi(\mathbf{z})$  will be the invariant solution of the combined operator of (3).

## 2.2 Constructing Samplers from Continuous and Jump Markov Processes

In Section 2.1, we described the evolution of the distribution  $p(\mathbf{z}|\mathbf{y}; t)$ . This evolution provides insight into the stationary distribution of the process. Here, we present the dynamics for an individual realization, which will play a critical role in our developed samplers.

A realization of the continuous Markov process,  $\frac{\partial}{\partial t} p(\mathbf{z}|\mathbf{y}; t) = \hat{\mathcal{L}} \left[ \frac{p(\mathbf{z}|\mathbf{y}; t)}{\pi(\mathbf{z})} \right]$ , can be generated from the stochastic differential equation (SDE):

$$d\mathbf{z} = \mathbf{f}(\mathbf{z})dt + \sqrt{2D(\mathbf{z})}d\mathbf{W}(t). \quad (6)$$

In practice, to simulate from (6), we consider an  $\epsilon$ -discretization:

$$\mathbf{z}_{t+1} \leftarrow \mathbf{z}_t + \epsilon_t \mathbf{f}(\mathbf{z}_t) + \eta_t, \quad \eta_t \sim \mathcal{N}(0, 2\epsilon_t D(\mathbf{z}_t)) \quad (7)$$

Although (7) is in the form of the Euler–Maruyama method, higher order numerical schemes can be used for better accuracy [Chen et al., 2015, Bou-Rabee and Owhadi, 2010, Leimkuhler et al., 2014]. The challenge here is to select  $\mathbf{f}$  and  $D$  such that (4) has the right stationary distribution, implying that the simulation from (6) provides samples from  $\pi(\mathbf{z})$ . Note that relying on a sample path from the discretized system of (7) typically leads to the introduction of bias due to discretization error. In these cases, the samples only provide unbiased estimates in the limit as  $\epsilon_t \rightarrow 0$  unless further corrections are introduced. In Section 3, we propose a reparameterization of (6) from which it is trivial to ensure the SDE has the desired stationary distribution. Then, in Section 5, we return to the idea of correcting for any potential discretization error if no bias can be tolerated.

Turning to the jump Markov process, the equation  $\frac{\partial}{\partial t} p(\mathbf{z}|\mathbf{y}; t) = \hat{\mathcal{J}} \left[ \frac{p(\mathbf{z}|\mathbf{y}; t)}{\pi(\mathbf{z})} \right]$  is approximately to the first order in  $\Delta t$  given by:

$$p(\mathbf{z}|\mathbf{y}; t + \Delta t) = \Delta t \int_{\mathbb{R}^d} W(\mathbf{z}|\mathbf{x}) p(\mathbf{x}|\mathbf{y}; t) d\mathbf{x} + \left[ 1 - \Delta t \int_{\mathbb{R}^d} W(\mathbf{x}|\mathbf{z}) d\mathbf{x} \right] p(\mathbf{z}|\mathbf{y}; t).$$

Although this is an approximation to the original equation, it has the same stationary distribution. Noting that  $p(\mathbf{x}|\mathbf{y}; 0) = \delta(\mathbf{z} - \mathbf{y})$ , we arise at the transition probability:

$$p(\mathbf{z}|\mathbf{y}; \Delta t) = \Delta t W(\mathbf{z}|\mathbf{y}) + \left[ 1 - \Delta t \int_{\mathbb{R}^d} W(\mathbf{x}|\mathbf{y}) d\mathbf{x} \right] \delta(\mathbf{z} - \mathbf{y}). \quad (8)$$

Equation (8) corresponds to a sampling process as follows. We take  $\Delta t$  to be the stepsize. Then, with probability  $\Delta t W(\mathbf{z}|\mathbf{y})$ , we transit from state  $\mathbf{y}$  to state  $\mathbf{z}$ . With probability  $1 - \Delta t \int_{\mathbb{R}^d} W(\mathbf{x}|\mathbf{y}) d\mathbf{x}$ , we stay in state  $\mathbf{y}$ .

Analogously to the challenge of selecting  $\mathbf{f}$  and  $D$ , the challenge here is to select the kernel  $W(\mathbf{z}|\mathbf{x})$  that leads to the stationary distribution being equal to the target distribution,  $\pi(\mathbf{z})$ . This requires that the positive transition kernel  $W$  satisfies  $\int_{\mathbb{R}^d} d\mathbf{x} [W(\mathbf{z}|\mathbf{x})\pi(\mathbf{x}) - W(\mathbf{x}|\mathbf{z})\pi(\mathbf{z})] = 0$ . Additionally, even if such a  $W$  can be defined, it might not define a distribution from which we can straightforwardly sample nor compute the necessary integral. Instead, in practice one typically resorts to implementing jump process samplers through the Metropolis-Hastings (MH) accept-reject scheme [Metropolis et al., 1953, Hastings, 1970]. In MH (Algorithm 1), one samples from a specified *proposal distribution*  $q(\mathbf{z}|\mathbf{y})$  and accepts the proposed value  $\mathbf{z}$  with probability

$$\alpha(\mathbf{y}, \mathbf{z}) = \min \left( 1, \frac{\pi(\mathbf{z})q(\mathbf{y}|\mathbf{z})}{\pi(\mathbf{y})q(\mathbf{z}|\mathbf{y})} \right). \quad (9)$$

In the form of (8), we have [Chib and Greenberg, 1995]:

$$p(\mathbf{z}|\mathbf{y}; \Delta t) = q(\mathbf{z}|\mathbf{y})\alpha(\mathbf{y}, \mathbf{z}) + \left[ 1 - \int_{\mathbb{R}^d} q(\mathbf{x}|\mathbf{y})\alpha(\mathbf{y}, \mathbf{x}) d\mathbf{x} \right] \delta(\mathbf{z} - \mathbf{y}). \quad (10)$$

When in state  $\mathbf{y}$  at time  $t$ , we propose to jump to state  $\mathbf{z}$  at  $t + \Delta t$  with conditional probability  $q(\mathbf{z}|\mathbf{y})$ , realized via a random number generator that has a distribution according to  $q(\mathbf{z}|\mathbf{y})$ ; we accept this proposal with probability  $\alpha(\mathbf{y}, \mathbf{z})$  to ensure that the target distribution will be preserved under this procedure. Hence, the total probability of transiting from state  $\mathbf{y}$  to  $\mathbf{z}$  is  $q(\mathbf{z}|\mathbf{y})\alpha(\mathbf{y}, \mathbf{z})$ . Otherwise, we stay in state  $\mathbf{y}$ .

Comparing (10) to (8), we see that MH restricts our attention to  $W(\mathbf{z}|\mathbf{y})$  satisfying  $\Delta t W(\mathbf{z}|\mathbf{y}) = q(\mathbf{z}|\mathbf{y})\alpha(\mathbf{y}, \mathbf{z})$ . We further see that the acceptance rate  $\alpha(\mathbf{y}, \mathbf{z})$  is specifically designed so that  $W(\mathbf{z}|\mathbf{x})\pi(\mathbf{x}) = W(\mathbf{x}|\mathbf{z})\pi(\mathbf{z})$ , a condition much stronger than  $\int_{\mathbb{R}^d} d\mathbf{x} [W(\mathbf{z}|\mathbf{x})\pi(\mathbf{x}) - W(\mathbf{x}|\mathbf{z})\pi(\mathbf{z})] = 0$ , in order to ensure that  $\pi(\mathbf{z})$  is the stationary distribution. However, this form restricts our attention solely to reversible processes. Instead, just as in the continuous Markov process case, in Section 4 we consider a reparameterization in terms of two kernel functions with straightforward-to-satisfy constraints. In that form, not only do we ensure the right stationary distribution, but we are able to devise a sampling algorithm for *irreversible* processes that has the same simplicity of implementation as MH.

As we will show in Sections 3 and 4, via a reparameterization of the operators  $\hat{\mathcal{L}}$  and  $\hat{\mathcal{J}}$ , we transform the problem of devising continuous and jump samplers with the right stationary distribution to one of simply specifying two matrices and two modestly constrained kernel functions. We can compose these two processes in various ways and still ensure that the overall sampler has the correct stationary distribution as is discussed in Section 5.

### 3 Continuous Markov Process Based Samplers

We start by examining how to use continuous processes—described by the diffusion operator  $\hat{\mathcal{L}}[\cdot]$  in (4) and realized by the SDE in (6)—to construct samplers. Although (6) provides a way to simulate the continuous

---

**Algorithm 1:** Metropolis-Hastings Algorithm

---

```

for  $t = 0, 1, 2 \dots N_{iter}$  do
    sample  $u \sim \mathcal{U}_{[0,1]}$ 
    propose  $\mathbf{z}(\ast) \sim q(\mathbf{z}(\ast)|\mathbf{z}(t))$ 
     $\alpha(\mathbf{z}^t, \mathbf{z}(\ast)) = \min \left\{ 1, \frac{\pi(\mathbf{z}(\ast)) q(\mathbf{z}(t)|\mathbf{z}(\ast))}{\pi(\mathbf{z}(t)) q(\mathbf{z}(\ast)|\mathbf{z}(t))} \right\}$ 
    if  $u < \alpha(\mathbf{z}(t), \mathbf{z}(\ast))$ ,  $\mathbf{z}(t+1) = \mathbf{z}(\ast)$ 
    else  $\mathbf{z}(t+1) = \mathbf{z}(t)$ 
end

```

---

dynamics and obtain samples from the Markov process, it is not clear which choice of  $\mathbf{f}$  and  $D$  will result in a stationary distribution equal to the target distribution from which we wish to sample. For a given  $\mathbf{f}$  and  $D$ , (4) allows us to analyze this stationary distribution, but it is very challenging to construct  $\mathbf{f}$  and  $D$  to yield a specified stationary distribution. Researchers have resorted to special cases such as overdamped Langevin [Roberts and Stramer, 2002, Welling and Teh, 2011], underdamped Langevin [Horowitz, 1991, Chen et al., 2014] and Nosé-Hoover [Ding et al., 2014, Shang et al., 2015] dynamics in the statistical physics literature for inspiration.

Instead, we propose to use an alternative form for the diffusion operator specified via two matrices  $D$  and  $Q$ , as well as the target distribution  $\pi(\mathbf{z})$  [Shi et al., 2012, Ma et al., 2015]:

$$\mathcal{L}[\varphi(\mathbf{z})] = \nabla^T \cdot \left( [D(\mathbf{z}) + Q(\mathbf{z})] [(\nabla \varphi(\mathbf{z})) \pi(\mathbf{z})] \right). \quad (11)$$

Here,  $D(\mathbf{z})$  is a positive semidefinite diffusion matrix and  $Q(\mathbf{z})$  a skew-symmetric matrix. An element-wise representation of  $\mathcal{L}[\varphi(\mathbf{z})]$  is:  $\sum_{i,j=1}^d \frac{\partial}{\partial \mathbf{z}_i} \left( [D_{ij}(\mathbf{z}) + Q_{ij}(\mathbf{z})] \left[ \left( \frac{\partial \varphi(\mathbf{z})}{\partial \mathbf{z}_j} \right) \pi(\mathbf{z}) \right] \right)$ . For this representation to be useful, we need to verify two properties: One is that  $\mathcal{L}[\cdot]$  has  $\pi(\mathbf{z})$  as its invariant distribution; the other is that any process with  $\pi(\mathbf{z})$  as the invariant distribution can be written in the form of (11) (i.e., there exists a  $D(\mathbf{z})$  and  $Q(\mathbf{z})$  that place the process in this representation). Together, these two properties (i) allow us to very straightforwardly explore a set of valid samplers by specifying pairs  $(D(\mathbf{z}), Q(\mathbf{z}))$  of positive semidefinite and skew-symmetric matrices, respectively, and (ii) ensure that as we span the space of all possible  $(D(\mathbf{z}), Q(\mathbf{z}))$ , we know we have covered all possible valid samplers. That is, our representation is *complete*.

It is straightforward to verify that  $p^s(\mathbf{z}) \propto \pi(\mathbf{z})$  is a stationary solution to the equation  $\frac{\partial p(\mathbf{z}|\mathbf{y};t)}{\partial t} = \mathcal{L} \left[ \frac{p(\mathbf{z}|\mathbf{y};t)}{\pi(\mathbf{z})} \right]$  with  $\mathcal{L}[\cdot]$  as in (11). More significantly, Theorem 1 states that any process with stationary distribution  $\pi(\mathbf{z})$  has a representation in our framework.

**Theorem 1.** Suppose  $\frac{\partial p(\mathbf{z}|\mathbf{y};t)}{\partial t} = \hat{\mathcal{L}} \left[ \frac{p(\mathbf{z}|\mathbf{y};t)}{\pi(\mathbf{z})} \right]$  for  $\hat{\mathcal{L}}[\cdot]$  as in (4) has stationary probability density function  $p^s(\mathbf{z}) \propto \pi(\mathbf{z})$ . Further assume that  $\left[ \mathbf{f}_i(\mathbf{z}) p^s(\mathbf{z}) - \sum_{j=1}^d \frac{\partial}{\partial \theta_j} \left( D_{ij}(\mathbf{z}) p^s(\mathbf{z}) \right) \right]$  is integrable with respect to the Lebesgue measure. Then, there exists a skew-symmetric matrix  $Q(\mathbf{z})$  such that the diffusion operator  $\hat{\mathcal{L}}[\cdot]$  is equivalent to the operator  $\mathcal{L}[\cdot]$  in (11).

We give a constructive proof of Theorem 1 in Appendix B. The equivalence between  $\mathcal{L}[\cdot]$  and  $\hat{\mathcal{L}}[\cdot]$  arises by taking

$$\mathbf{f}(\mathbf{z}) = -[D(\mathbf{z}) + Q(\mathbf{z})] \nabla H(\mathbf{z}) + \Gamma(\mathbf{z}),$$

for  $H(\mathbf{z}) = -\log(\pi(\mathbf{z}))$  and  $\Gamma_i(\mathbf{z}) = \sum_{j=1}^d \frac{\partial}{\partial \mathbf{z}_j} (D_{ij}(\mathbf{z}) + Q_{ij}(\mathbf{z}))$ . Hence, the SDE realization of the Markov processes following the new  $\mathcal{L}$  operator is

$$d\mathbf{z} = \left[ - (D(\mathbf{z}) + Q(\mathbf{z})) \nabla H(\mathbf{z}) + \Gamma(\mathbf{z}) \right] dt + \sqrt{2D(\mathbf{z})} d\mathbf{W}(t). \quad (12)$$

### 3.1 Decomposing $\mathcal{L}[\cdot]$ into Reversible and Irreversible Dynamics

The operator  $\mathcal{L}[\cdot]$  can be decomposed into a symmetric part  $\mathcal{L}^S[\cdot]$ , characterizing a reversible Markov process, and an anti-symmetric part  $\mathcal{L}^A[\cdot]$ , representing an irreversible process. The symmetric and skew-symmetric operators corresponds to two different kinds of dynamics.

The symmetric operator is determined solely by the diffusion matrix  $D$  since the skew-symmetric matrix  $Q$  cancels out:

$$\begin{aligned} \mathcal{L}^S \left[ \frac{p(\mathbf{z}|\mathbf{y}; t)}{\pi(\mathbf{z})} \right] &= \frac{1}{2} \left( \mathcal{L} \left[ \frac{p(\mathbf{z}|\mathbf{y}; t)}{\pi(\mathbf{z})} \right] + \mathcal{L}^* \left[ \frac{p(\mathbf{z}|\mathbf{y}; t)}{\pi(\mathbf{z})} \right] \right) = \nabla^T \cdot \left( D(\mathbf{z}) \left[ \nabla \left( \frac{p(\mathbf{z}|\mathbf{y}; t)}{\pi(\mathbf{z})} \right) \pi(\mathbf{z}) \right] \right) \\ &= \sum_{i,j=1}^d \frac{\partial^2}{\partial \mathbf{z}_i \partial \mathbf{z}_j} \left( D_{ij}(\mathbf{z}) p(\mathbf{z}|\mathbf{y}; t) \right) + \sum_{i=1}^d \frac{\partial}{\partial \mathbf{z}_i} \left( \left[ \sum_j D_{ij}(\mathbf{z}) \frac{\partial H(\mathbf{z})}{\partial \mathbf{z}_j} - \Gamma_i^D(\mathbf{z}) \right] p(\mathbf{z}|\mathbf{y}; t) \right). \end{aligned} \quad (13)$$

Here,  $\Gamma_i^D(\mathbf{z}) = \sum_{j=1}^d \frac{\partial}{\partial \mathbf{z}_j} D_{ij}(\mathbf{z})$  and  $\mathcal{L}^*[\cdot]$  denotes the adjoint operator of  $\mathcal{L}[\cdot]$ . According to Itô's convention, the last two lines of (13) imply that  $\frac{\partial}{\partial t} p(\mathbf{z}|\mathbf{y}; t) = \mathcal{L}^S \left[ \frac{p(\mathbf{z}|\mathbf{y}; t)}{\pi(\mathbf{z})} \right]$  corresponds to reversible Brownian motion in a potential force field on a Riemannian manifold specified by the diffusion matrix  $D(\mathbf{z})$ :  $d\mathbf{z} = \left[ -D(\mathbf{z}) \nabla H(\mathbf{z}) + \Gamma^D(\mathbf{z}) \right] dt + \sqrt{2D(\mathbf{z})} d\mathbf{W}(t)$ . This is referred to as *Riemannian Langevin dynamics* [Roberts and Stramer, 2002]. When  $D(\mathbf{z})$  is positive definite, the reversible Markov dynamics have nice statistical regularity and will drive the system to converge to the stationary distribution.

The anti-symmetric operator is dictated solely by  $Q$ , as here  $D$  cancels out:

$$\begin{aligned} \mathcal{L}^A \left[ \frac{p(\mathbf{z}|\mathbf{y}; t)}{\pi(\mathbf{z})} \right] &= \frac{1}{2} \left( \mathcal{L} \left[ \frac{p(\mathbf{z}|\mathbf{y}; t)}{\pi(\mathbf{z})} \right] - \mathcal{L}^* \left[ \frac{p(\mathbf{z}|\mathbf{y}; t)}{\pi(\mathbf{z})} \right] \right) = \nabla^T \cdot \left( Q(\mathbf{z}) \left[ \nabla \left( \frac{p(\mathbf{z}|\mathbf{y}; t)}{\pi(\mathbf{z})} \right) \pi(\mathbf{z}) \right] \right) \\ &= \nabla^T \cdot \left( \left[ Q(\mathbf{z}) \nabla H(\mathbf{z}) - \Gamma^Q(\mathbf{z}) \right] p(\mathbf{z}|\mathbf{y}; t) \right). \end{aligned} \quad (14)$$

Here,  $\Gamma_i^Q(\mathbf{z}) = \sum_{j=1}^d \frac{\partial}{\partial \mathbf{z}_j} Q_{ij}(\mathbf{z})$ . The last line of (14) demonstrates that  $\frac{\partial}{\partial t} p(\mathbf{z}|\mathbf{y}; t) = \mathcal{L}^A \left[ \frac{p(\mathbf{z}|\mathbf{y}; t)}{\pi(\mathbf{z})} \right]$  is a Liouville equation, which describes the density evolution of  $p(\mathbf{z}|\mathbf{y}; t)$  according to conserved, deterministic dynamics:  $d\mathbf{z}/dt = -Q(\mathbf{z}) \nabla H(\mathbf{z}) + \Gamma^Q(\mathbf{z})$ , with  $\pi(\mathbf{z})$  its invariant measure.

Combining the dynamics of (13) and (14) leads to a general SDE, (12), with stationary distribution  $\pi(\mathbf{z})$ . Previous methods [Roberts and Stramer, 2002, Xifara et al., 2014, Welling and Teh, 2011, Patterson and Teh, 2013, Neal, 2010] have primarily focused on solely symmetric or anti-symmetric operators,  $\mathcal{L}^S$  or  $\mathcal{L}^A$ , respectively, as we make explicit in Section 3.3. In [Ma et al., 2015], we have described some samplers using both operators, but without a general procedure for constructing and analyzing such samplers.



---

**Algorithm 2:** Continuous Markov Process Sampling Algorithm

---

```

initialize  $\mathbf{z}_0$ 
for  $t = 0, 1, 2 \dots N_{iter}$  do
  for  $i = 1 \dots n$  do
     $\Gamma_i(\mathbf{z}) = \sum_j \frac{\partial}{\partial \mathbf{z}_j} (D_{ij}(\mathbf{z}) + Q_{ij}(\mathbf{z}))$ 
  end
  sample  $\eta_t \sim \mathcal{N}(0, 2\epsilon_t D(\mathbf{z}_t))$ 
   $\mathbf{z}_{t+1} \leftarrow \mathbf{z}_t - \epsilon_t [(D(\mathbf{z}_t) + Q(\mathbf{z}_t)) \nabla H(\mathbf{z}_t) + \Gamma(\mathbf{z}_t)] + \eta_t$ 
end

```

---

### 3.2 Continuous Markov Process Sampling Algorithm

Following (7), we can simulate from the SDE underlying the operator in (12) using the following  $\epsilon$ -discretization:

$$\mathbf{z}_{t+1} \leftarrow \mathbf{z}_t - \epsilon_t [(D(\mathbf{z}_t) + Q(\mathbf{z}_t)) \nabla H(\mathbf{z}_t) + \Gamma(\mathbf{z}_t)] + \eta_t, \quad \eta_t \sim \mathcal{N}(0, 2\epsilon_t D(\mathbf{z}_t)). \quad (15)$$

Again, higher-order numerical schemes can be used in place of the first-order integrator above [Chen et al., 2015, Bou-Rabee and Owhadi, 2010, Leimkuhler et al., 2014]. The resulting algorithm is outlined in Algorithm 2. (Recall that bias is introduced via the discretization, i.e., setting  $\epsilon_t$  finite. We will return to this in Section 5.)

### 3.3 Previous MCMC Algorithms as Special Cases

We explicitly state how some previous continuous-dynamic-based MCMC methods fit within the proposed framework based on specific choices of  $D(\mathbf{z})$ ,  $Q(\mathbf{z})$  and  $H(\mathbf{z})$ . We show how our framework can be used to “reinvent” the samplers by guiding their construction and avoiding potential mistakes or inefficiencies caused by naïve implementations.

**Hamiltonian Monte Carlo (HMC)** The key ingredient in HMC [Duane et al., 1987, Neal, 2010] is Hamiltonian dynamics, which simulates the physical motion of an object with position  $\theta$ , momentum  $r$ , and mass  $M$  on an frictionless surface as follows:

$$\begin{cases} \theta_{t+1} \leftarrow \theta_t + \epsilon_t M^{-1} r_t \\ r_{t+1} \leftarrow r_t - \epsilon_t \nabla U(\theta_t). \end{cases} \quad (16)$$

(Note that above we consider a straightforward integrator, whereas in practice it is common to use a leapfrog simulation instead [Neal, 2010].) Equation (16) is a special case of the proposed framework with  $\mathbf{z} = (\theta, r)$ ,  $H(\theta, r) = U(\theta) + \frac{1}{2} r^T M^{-1} r$ ,  $Q(\theta, r) = \begin{pmatrix} 0 & -I \\ I & 0 \end{pmatrix}$  and  $D(\theta, r) = \mathbf{0}$ .

**Langevin Dynamics** The Langevin dynamics sampler [Roberts and Stramer, 2002, Welling and Teh, 2011] proposes to use the following first order (no momentum) Langevin dynamics to generate samples

$$\theta_{t+1} \leftarrow \theta_t - \epsilon_t D \nabla U(\theta_t) + \eta_t, \quad \eta_t \sim \mathcal{N}(0, 2\epsilon_t D). \quad (17)$$

This algorithm corresponds to taking  $\mathbf{z} = \theta$  with  $H(\theta) = U(\theta)$ ,  $D(\theta) = D$ , and  $Q(\theta) = \mathbf{0}$ .



**Riemannian Langevin Dynamics** The Langevin dynamics sampler can be generalized to use an adaptive diffusion matrix  $D(\theta)$ . Specifically, it is interesting to take  $D(\theta) = G^{-1}(\theta)$ , where  $G(\theta)$  is the Fisher information metric [Xifara et al., 2014, Patterson and Teh, 2013]. The sampler iterates

$$\theta_{t+1} \leftarrow \theta_t - \epsilon_t [G(\theta_t)^{-1} \nabla U(\theta_t) + \Gamma(\theta_t)] + \eta_t, \quad \eta_t \sim \mathcal{N}(0, 2\epsilon_t G(\theta_t)^{-1}). \quad (18)$$

We can cast this Riemannian Langevin dynamics sampler [Patterson and Teh, 2013] into our framework taking  $D(\theta) = G(\theta)^{-1}$ , and  $Q(\theta) = \mathbf{0}$ . From our framework, we know that here  $\Gamma_i(\theta) = \sum_j \frac{\partial D_{ij}(\theta)}{\partial \theta_j}$ . Interestingly, in

earlier literature [Girolami and Calderhead, 2011],  $\Gamma_i(\theta)$  was taken to be  $2 |G(\theta)|^{-1/2} \sum_j \frac{\partial}{\partial \theta_j} (G_{ij}^{-1}(\theta) |G(\theta)|^{1/2})$ .

More recently, it was found that this correction term corresponds to the distribution function with respect to a non-Lebesgue measure [Roberts and Stramer, 2002]. For the Lebesgue measure, the revised  $\Gamma_i(\theta)$  was as determined by our framework [Roberts and Stramer, 2002]. This is an example of how our theory provides guidance in devising correct samplers.

**Summary of past samplers** In our framework, the Langevin dynamic based samplers take  $Q(\mathbf{z}) = 0$  and instead stress the design of the diffusion matrix  $D(\mathbf{z})$ . The standard Langevin dynamic sampler uses a constant  $D(\mathbf{z})$ , whereas the Riemannian variant uses an adaptive,  $\mathbf{z}$ -dependent diffusion matrix to better account for the geometry of the space being explored. On the other hand, HMC takes  $D(\mathbf{z}) = 0$  and focuses on the curl matrix  $Q(\mathbf{z})$ . As we see, our method is a generalization of the dynamics underlying Hamiltonian Monte Carlo (and Riemannian Hamiltonian Monte Carlo) methods, extending the symplectic structure to be non-constant. That is, considering general skew-symmetric matrices instead of just  $Q(\mathbf{z}) = \begin{pmatrix} 0 & -I \\ I & 0 \end{pmatrix}$  as in [Tuckerman et al., 2001, Fang et al., 2014]. The generalized Hamiltonian dynamics can explore the state space rapidly, and are guaranteed to preserve the stationary distribution once it is achieved.

Examination of the past methods provides us with the insight that  $D(\mathbf{z})$  can enable diffusive exploration across local modes. And just as in HMC,  $Q(\mathbf{z})$  drives the sampler to walk along contours of equal probability allowing it to rapidly traverse regions of lower probability, especially when state adaptation is incorporated. Importantly, through our  $(D(\mathbf{z}), Q(\mathbf{z}))$  parameterization, we can readily examine which parts of the product space  $D(\mathbf{z}) \times Q(\mathbf{z})$ —representing the space of all possible samplers—have been covered. We see that a majority of possible samplers have not yet been considered. For ways in which to use our framework to construct new samplers, see [Ma et al., 2015].

## 4 Samplers Using Jump Processes with Operator $\mathcal{J}[\cdot]$

We now turn our attention to the jump operator  $\hat{\mathcal{J}}[\cdot]$  of (5). As with the continuous dynamic operator  $\hat{\mathcal{L}}[\cdot]$ , we can consider an equivalent representation that enables more ready analysis of the properties of the process, and the development of efficient irreversible jump process samplers. Based on the form of (8) specified by generic kernel  $W(\mathbf{x}|\mathbf{z})$ , it is challenging to determine which choice of  $W(\mathbf{x}|\mathbf{z})$  leads to a jump process with the correct stationary distribution. Even if one can construct such a  $W$ , it can be challenging to use  $W$  to sample a realization of the jump process; instead, often one restricts attention to reversible processes and uses MH (see Section 2.2). We instead consider an equivalent but alternative representation defined in terms of two kernels  $S$  and  $A$ . A simple set of constraints on  $S$  and  $A$  ensures that the target distribution  $\pi(\mathbf{z})$  is the stationary distribution of the jump process.

In particular, we consider

$$\mathcal{J}[\varphi(\mathbf{z})] = \int_{\mathbb{R}^d} d\mathbf{x} [(S(\mathbf{x}, \mathbf{z}) + A(\mathbf{x}, \mathbf{z}))\varphi(\mathbf{x}) - S(\mathbf{x}, \mathbf{z})\varphi(\mathbf{z})], \quad (19)$$

where  $S$  is a symmetric kernel and  $A$  is an anti-symmetric kernel. Based on the form of (19), as shown in Appendix C.1, we simply have to satisfy the following constraints in order to ensure that  $\pi(\mathbf{z})$  is the stationary distribution:

1.  $\int_{\mathbb{R}^d} S(\mathbf{x}, \mathbf{z}) \pi^{-1}(\mathbf{x}) d\mathbf{x}$  and  $\int_{\mathbb{R}^d} A(\mathbf{x}, \mathbf{z}) \pi^{-1}(\mathbf{x}) d\mathbf{x}$  exist
2.  $S(\mathbf{x}, \mathbf{z}) + A(\mathbf{x}, \mathbf{z}) > 0$
3.  $\int_{\mathbb{R}^d} A(\mathbf{x}, \mathbf{z}) d\mathbf{x} = 0$ .

Following (8), the transition probability implied by the operator of (19) assuming a  $\Delta t$ -discretization is given by:

$$p(\mathbf{z}|\mathbf{y}; \Delta t) = \frac{\Delta t}{\pi(\mathbf{y})} (S(\mathbf{y}, \mathbf{z}) + A(\mathbf{y}, \mathbf{z})) + \left[ 1 - \frac{\Delta t}{\pi(\mathbf{y})} \int_{\mathbb{R}^d} S(\mathbf{y}, \mathbf{x}) d\mathbf{x} \right] \delta(\mathbf{z} - \mathbf{y}). \quad (20)$$

Since the jump operator has  $\pi(\mathbf{z})$  as the stationary distribution assuming the constraints of  $S$  and  $A$  are satisfied, the transition probability of (20) defines a valid procedure for drawing samples from the target  $\pi(\mathbf{z})$ . In particular, over time  $\Delta t$ , state  $\mathbf{y}$  transitions to state  $\mathbf{z}$  with probability  $\Delta t(S(\mathbf{y}, \mathbf{z}) + A(\mathbf{y}, \mathbf{z}))/\pi(\mathbf{y})$ , and state  $\mathbf{y}$  remains unchanged with probability  $\left[ 1 - \frac{\Delta t}{\pi(\mathbf{y})} \int_{\mathbb{R}^d} S(\mathbf{y}, \mathbf{x}) d\mathbf{x} \right]$ . In Section 4.3, we examine a practical algorithm for efficiently implementing such a procedure based on an accept-reject scheme analogous to the MH algorithm outlined in Algorithm 1. The important challenge we conquer is handling the irreversibility of the process arising from  $A \neq 0$ . We first study these irreversible dynamics in Section 4.1.

#### 4.1 Reversible and Irreversible Dynamics of $\mathcal{J}[\cdot]$

Similar to operator  $\mathcal{L}[\cdot]$ , operator  $\mathcal{J}[\cdot]$  can also be decomposed into a symmetric (reversible) part  $\mathcal{J}^S[\cdot]$  and anti-symmetric (irreversible) part  $\mathcal{J}^A[\cdot]$ :

$$\mathcal{J}^S[\varphi(\mathbf{z})] = \frac{1}{2} (\mathcal{J}[\varphi(\mathbf{z})] + \mathcal{J}^*[\varphi(\mathbf{z})]) = \int_{\mathbb{R}^d} d\mathbf{x} [S(\mathbf{x}, \mathbf{z})\varphi(\mathbf{x}) - S(\mathbf{x}, \mathbf{z})\varphi(\mathbf{z})]; \quad (21)$$

$$\mathcal{J}^A[\varphi(\mathbf{z})] = \frac{1}{2} (\mathcal{J}[\varphi(\mathbf{z})] - \mathcal{J}^*[\varphi(\mathbf{z})]) = \int_{\mathbb{R}^d} d\mathbf{x} [A(\mathbf{x}, \mathbf{z})\varphi(\mathbf{x})]. \quad (22)$$

Here,  $\mathcal{J}^*[\cdot]$  is the adjoint operator of  $\mathcal{J}[\cdot]$ . We see that  $A$  fully determines the irreversible dynamics whereas  $S$  defines the reversible part. We can further derive from (20) that  $A$  is the difference between the probability of a forward path and the backward path:

$$A(\mathbf{x}, \mathbf{z}) = \frac{1}{2\Delta t} (\pi(\mathbf{y})p(\mathbf{z}|\mathbf{y}; \Delta t) - \pi(\mathbf{z})p(\mathbf{y}|\mathbf{z}; \Delta t)). \quad (23)$$

#### 4.2 Previous Samplers as Special Cases

As with past continuous-dynamic-based samplers, we now cast a set of past jump-process-based samplers into our framework.

**Direct resampling** Methods that sample directly from  $\pi(\mathbf{z})$  take  $S(\mathbf{y}, \mathbf{z}) = \frac{1}{\Delta t} \pi(\mathbf{y})\pi(\mathbf{z})$  and  $A(\mathbf{y}, \mathbf{z}) = 0$ . We can verify this by substituting into (20).

**Metropolis-Hastings** The very popular MH algorithm falls into our framework taking  $A(\mathbf{y}, \mathbf{z}) = 0$  and

$$S(\mathbf{y}, \mathbf{z}) = \frac{1}{\Delta t} \min(\pi(\mathbf{y})q(\mathbf{z}|\mathbf{y}), \pi(\mathbf{z})q(\mathbf{y}|\mathbf{z})). \quad (24)$$

To see this, we refer to Section 2.2. The specified form for  $S$  and  $A$  above arises from comparing the transition probability of (10) with that of (20).

**Summary of past samplers** In the previously mentioned algorithms, and a majority of those used in practice, only reversible Markov jump processes ( $A(\mathbf{z}, \mathbf{y}) = 0$ ) are considered. In Section 4.3, we explore the case where the process is irreversible, i.e.,  $A(\mathbf{z}, \mathbf{y}) \neq 0$ .

### 4.3 Construction of Irreversible Jump Processes for MCMC

Analogous to the discussion of Section 2.2, there are two issues of designing samplers with Markov jump processes. One is the construction of transition kernels, a task that has been alleviated in part by the new formulation of (20) in terms of  $S(\mathbf{y}, \mathbf{z})$  and  $A(\mathbf{y}, \mathbf{z})$  with simple constraints, though we still have to construct such kernels. Another is simulating the Markov process of (20). In all but the simplest cases, we might not be able to sample from the transition probability  $\Delta t \cdot (S(\mathbf{y}, \mathbf{z}) + A(\mathbf{y}, \mathbf{z}))/\pi(\mathbf{y})$ . These two issues are often intertwined posing challenges to the design of samplers. As mentioned in Section 2.2, the MH algorithm is often resorted to due to its ease of implementation. It separates the process of proposing a sample into two simple steps: (1) proposing a candidate according to a known conditional probability distribution  $q(\mathbf{z}|\mathbf{y})$  and (2) accepting or rejecting the candidate according to a certain probability. An important drawback of the vanilla MH sampler, however, is that the reversibility of the jump process being designed can greatly restrict possible ways to increase the mixing of the Markov chain.

There have been previous efforts to break the restriction of reversibility in different cases. For example, the *non-reversible MH* algorithm adds a vorticity function to the MH procedure [Bierkens, 2015] while the *lifting method* makes two replica of the original state space with a skew detailed balance condition to facilitate irreversibility [Turitsyn et al., 2011, Vucelja, 2015]. The authors have shown examples of sampling special distributions, but it is unclear how to generalize these past methods to handle a broad set of target distributions. See Section 6 for a detailed discussion of these and other methods. Here, we show how we can devise a practical and efficient irreversible jump process algorithm analogous to MH that can be applied to general targets; this procedure implicitly defines valid kernels  $S(\mathbf{y}, \mathbf{z})$  and  $A(\mathbf{y}, \mathbf{z})$ . In particular, just as MH corresponds to restricting the class of kernels  $W(\mathbf{z}|\mathbf{y})$ , our algorithm also focuses in on particular instances of  $A(\mathbf{y}, \mathbf{z})$ , but importantly allows  $A(\mathbf{y}, \mathbf{z}) \neq 0$  (i.e., irreversible processes). The value of this in practice is demonstrated in the experiments of Section 7.

**A naïve approach** A straightforward approach to revise the MH algorithm to make antisymmetric kernel  $A(\mathbf{y}, \mathbf{z})$  nonzero, resulting in an irreversible sampler, is to utilize different proposal distributions  $f(\mathbf{z}|\mathbf{y})$  and  $g(\mathbf{z}|\mathbf{y})$ , instead of a single  $q(\mathbf{z}|\mathbf{y})$ . That is, the transition kernel of the MH algorithm in (24) is changed to

$$F(\mathbf{y}, \mathbf{z}) = S(\mathbf{y}, \mathbf{z}) + A(\mathbf{y}, \mathbf{z}) = \frac{1}{\Delta t} \min(\pi(\mathbf{y})f(\mathbf{z}|\mathbf{y}), \pi(\mathbf{z})g(\mathbf{y}|\mathbf{z})). \quad (25)$$

Here we are considering jump processes with  $A(\mathbf{y}, \mathbf{z}) = \frac{1}{2}(F(\mathbf{y}, \mathbf{z}) - F(\mathbf{z}, \mathbf{y})) \neq 0$ , in contrast to what we saw for MH. By adjusting  $f$  and  $g$ , faster mixing rates can possibly be attained while maintaining a simple sampling procedure akin to that of MH (see Algorithm 1, but with  $f$  in place of  $q$  in the numerator and  $g$  in place of  $q$  in the denominator in the  $\alpha$  calculation). The more  $f$  and  $g$  differ, the more irreversibility effect is incorporated in the design of the sampler. Functions  $f$  and  $g$  can even be selected to have non-symmetric support in the state space (as is chosen in our experiments), so that new proposals are guided in certain directions until being

---

**Algorithm 3: One-Dimensional Irreversible Jump Sampler**


---

```

randomly pick  $z^p$  from  $\{1, -1\}$  with equal probability
for  $t = 0, 1, 2 \dots N_{iter}$  do
  sample  $u \sim \mathcal{U}_{[0,1]}$ 
  if  $z^p > 0$  then
    sample  $\mathbf{z}(\ast) \sim f(\mathbf{z}(\ast)|\mathbf{z}(t))$ 
     $\alpha(\mathbf{z}(t), \mathbf{z}(\ast)) = \min \left\{ 1, \frac{\pi(\mathbf{z}(\ast)) g(\mathbf{z}(t)|\mathbf{z}(\ast))}{\pi(\mathbf{z}(t)) f(\mathbf{z}(\ast)|\mathbf{z}(t))} \right\}$ 
  end
  else
    sample  $\mathbf{z}(\ast) \sim g(\mathbf{z}(\ast)|\mathbf{z}(t))$ 
     $\alpha(\mathbf{z}(t), \mathbf{z}(\ast)) = \min \left\{ 1, \frac{\pi(\mathbf{z}(\ast)) f(\mathbf{z}(t)|\mathbf{z}(\ast))}{\pi(\mathbf{z}(t)) g(\mathbf{z}(\ast)|\mathbf{z}(t))} \right\}$ 
  end
  if  $u < \alpha(\mathbf{z}(t), \mathbf{z}(\ast))$ ,  $\mathbf{z}(t+1) = \mathbf{z}(\ast)$ ;  $z^p(t+1) = z^p(t)$ 
  else  $\mathbf{z}(t+1) = \mathbf{z}(t)$ ;  $z^p(t+1) = -z^p(t)$ 
end

```

---

rejected, encouraging the algorithm to explore farther states. The primary issue with this construction is that  $\int_{\mathbb{R}^d} A(\mathbf{y}, \mathbf{z}) d\mathbf{y} \neq 0$  in general, rendering the stationary distribution *not* the  $\pi(\mathbf{z})$  that we desire. The question is how to design the anti-symmetric kernel  $A(\mathbf{y}, \mathbf{z})$ , such that  $\int_{\mathbb{R}^d} A(\mathbf{y}, \mathbf{z}) d\mathbf{y} = 0$ .

**Lifting for sampling when  $d = 1$**  A simple modified approach is to follow an adjoint Markov process after being rejected by the original one. This is inspired by the *lifting* idea in discrete spaces [Turitsyn et al., 2011, Vucelja, 2015]. Importantly, this approach has  $\pi(\mathbf{z})$  as the stationary distribution.

Algorithmically, this process introduces a one-dimensional, uniformly distributed discrete auxiliary variable  $\mathbf{y}^p \in \{-1, 1\}$ . We then define

$$\begin{aligned} \tilde{f}(\mathbf{z}, \mathbf{z}^p | \mathbf{y}, \mathbf{y}^p) &= (\mathcal{H}(\mathbf{y}^p) f(\mathbf{z} | \mathbf{y}) + \mathcal{H}(-\mathbf{y}^p) g(\mathbf{z} | \mathbf{y})) \\ \tilde{g}(\mathbf{z}, \mathbf{z}^p | \mathbf{y}, \mathbf{y}^p) &= (\mathcal{H}(-\mathbf{y}^p) f(\mathbf{z} | \mathbf{y}) + \mathcal{H}(\mathbf{y}^p) g(\mathbf{z} | \mathbf{y})), \end{aligned} \quad (26)$$

where  $f(\mathbf{z} | \mathbf{y})$  and  $g(\mathbf{z} | \mathbf{y})$  are different conditional probability distributions, and  $\mathcal{H}$  is the Heaviside function:

$$\mathcal{H}(\mathbf{y}^p) = \begin{cases} 1 & \mathbf{y}^p \geq 0 \\ 0 & \mathbf{y}^p < 0. \end{cases}$$

We modify the MH algorithm as described in Algorithm 3, where we update state  $\mathbf{y}$  and the auxiliary variable  $\mathbf{y}^p$  according to the following transition probability (as in our recipe of (20)):

$$\begin{aligned} p(\mathbf{z}, \mathbf{z}^p | \mathbf{y}, \mathbf{y}^p; \Delta t) &= \frac{\Delta t}{\pi(\mathbf{y})\pi(\mathbf{y}^p)} \delta(\mathbf{z}^p - \mathbf{y}^p) \cdot \mathfrak{F}(\mathbf{y}, \mathbf{y}^p, \mathbf{z}, \mathbf{z}^p) \\ &\quad + \delta(\mathbf{z}^p + \mathbf{y}^p) \delta(\mathbf{z} - \mathbf{y}) \left( 1 - \frac{\Delta t}{\pi(\mathbf{y})\pi(\mathbf{y}^p)} \int_{\mathbb{R}^d} \mathfrak{F}(\mathbf{y}, \mathbf{y}^p, \mathbf{x}, -\mathbf{z}^p) d\mathbf{x} \right), \end{aligned} \quad (27)$$

in which  $\mathfrak{F}(\mathbf{y}, \mathbf{y}^p, \mathbf{z}, \mathbf{z}^p)$  is defined using  $\tilde{f}$  and  $\tilde{g}$ :

$$\mathfrak{F}(\mathbf{y}, \mathbf{y}^p, \mathbf{z}, \mathbf{z}^p) = \min \left( \pi(\mathbf{y})\pi(\mathbf{y}^p) \tilde{f}(\mathbf{z}, \mathbf{z}^p | \mathbf{y}, \mathbf{y}^p), \pi(\mathbf{z})\pi(\mathbf{z}^p) \tilde{g}(\mathbf{y}, \mathbf{y}^p | \mathbf{z}, \mathbf{z}^p) \right).$$

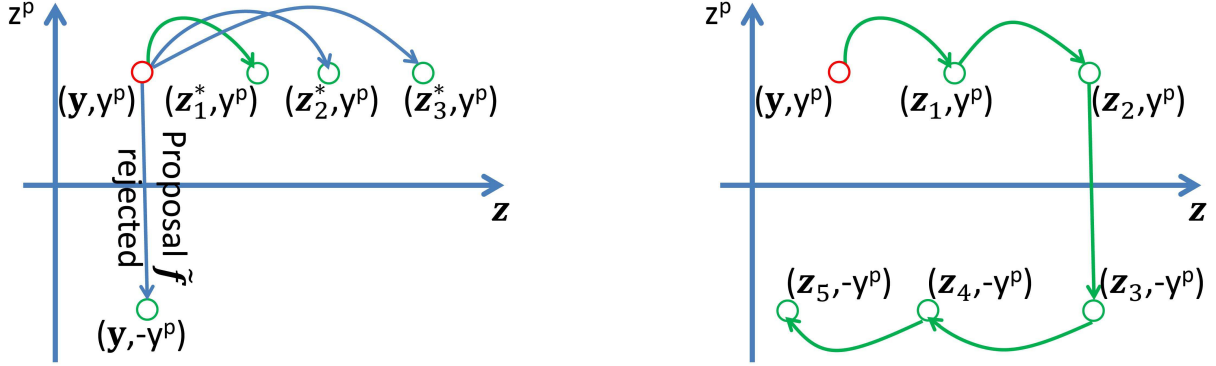


Figure 1: Update rule starting from state  $(y, y^p)$ . *Left:* Several possible states  $(z^*, z^p)$  that the algorithm could visit in the next step. Without resampling the auxiliary variables,  $z^p$  can only be  $y^p$  or  $-y^p$ . *Right:* Assuming the algorithm visits  $(z_1, y^p)$  as the next state to  $(y, y^p)$  (indicated by the green arrow), a sample trajectory of states generated.

This update rule can be understood as follows. With probability  $\mathfrak{F}(y, y^p, z, z^p)/(\pi(y)\pi(y^p))$ , state  $y$  becomes state  $z$  while the auxiliary state  $y^p$  remains the same. Alternatively, with probability

$\left[1 - \frac{1}{\pi(y)\pi(y^p)} \int_{\mathbb{R}^d} \mathfrak{F}(y, y^p, x, -z^p) dx\right]$ , no new state  $(x, y^p)$  is accepted conditioning on currently being at state  $(y, y^p)$ . Instead, state  $(y, y^p)$  is directly changed to state  $(y, -y^p)$ , leading to a different jump process in  $y$ . An illustration of the update rule is shown in Fig. 1.

From (23), we see that this proposed algorithm takes the anti-symmetric kernel  $A(y, y^p, z, z^p)$  to be

$$A(y, y^p, z, z^p) = \frac{1}{2\Delta t} \left( \pi(y)\pi(y^p)p(z, z^p|y, y^p; \Delta t) - \pi(z)\pi(z^p)p(y, y^p|z, z^p; \Delta t) \right) \quad (28)$$

with  $p(z, z^p|y, y^p; \Delta t)$  as in (27). To ensure correctness of the sampler,  $A(y, y^p, z, z^p)$  must satisfy (condition 3):

$$\int_{\mathbb{R}^{d+1}} A(y, y^p, z, z^p) dy dy^p = 0.$$

In Appendix C.2, we prove that this is indeed the case for  $A(y, y^p, z, z^p)$  as in (28). The intuition is that the jump in the auxiliary variable introduces a circulative behavior to the whole process (see Fig. 1 for illustration). This circulation of probability flux is exactly balanced with the jumps in the original variable and the auxiliary variable. We also see in Fig. 1 that irreversibility introduces a directional effect (just like HMC introduces a direction of rotation). This algorithm is a generalization of the *guided walk Metropolis* method [Gustafson, 1998] and works well in one dimension, as we demonstrate in Section 7.2. In what follows, we generalize this idea to higher dimensions  $d > 1$ .

**Moving to higher dimensions** An irreversible sampler in  $\mathbb{R}^d$  can be constructed as follows. We expand the state space by introducing a  $d^p$ -dimensional auxiliary variable  $y^p \in \mathbb{R}^{d^p}$  in the new state space  $(y, y^p)$ . The total probability can be designated as:  $\pi(y, y^p) = \pi(y)\pi(y^p)$ . We further impose symmetry on the auxiliary variables

---

**Algorithm 4:** Monte Carlo Algorithm from Irreversible Jump Process

---

```

for  $t = 0, 1, 2 \dots N_{iter}$  do
    optionally, periodically resample auxiliary variable  $\mathbf{z}^p$  as  $\mathbf{z}^p(t) \sim \pi(\mathbf{z}^p)$ 
    sample  $u \sim \mathcal{U}_{[0,1]}$ 
    sample  $\mathbf{z}(\ast) \sim \tilde{f}(\mathbf{z}(\ast), \mathbf{z}^p(\ast) | \mathbf{z}(t), \mathbf{z}^p(t))$ 
     $\alpha(\mathbf{z}(t), \mathbf{z}^p(t), \mathbf{z}(\ast), \mathbf{z}^p(\ast)) = \min \left\{ 1, \frac{\pi(\mathbf{z}(\ast)) \pi(\mathbf{z}^p(\ast)) \tilde{g}(\mathbf{z}(t), \mathbf{z}^p(t) | \mathbf{z}(\ast), \mathbf{z}^p(\ast))}{\pi(\mathbf{z}(t)) \pi(\mathbf{z}^p(t)) \tilde{f}(\mathbf{z}(\ast), \mathbf{z}^p(\ast) | \mathbf{z}(t), \mathbf{z}^p(t))} \right\}$ 
    if  $u < \alpha(\mathbf{z}(t), \mathbf{z}^p(t), \mathbf{z}(\ast), \mathbf{z}^p(\ast))$ ,  $(\mathbf{z}(t+1), \mathbf{z}^p(t+1)) = (\mathbf{z}(\ast), \mathbf{z}^p(\ast))$ 
    else  $(\mathbf{z}(t+1), \mathbf{z}^p(t+1)) = (\mathbf{z}(t), -\mathbf{z}^p(t))$ 
end

```

---

such that  $\pi(\mathbf{y}^p) = \pi(-\mathbf{y}^p)$ , and let

$$\begin{aligned}
 \tilde{f}(\mathbf{z}, \mathbf{z}^p | \mathbf{y}, \mathbf{y}^p) &= \prod_{i=1}^{d^p} (\mathcal{H}(\mathbf{y}_i^p) f_i(\mathbf{z} | \mathbf{y}, \mathbf{y}_i^p) + \mathcal{H}(-\mathbf{y}_i^p) g_i(\mathbf{z} | \mathbf{y}, -\mathbf{y}_i^p)); \\
 \tilde{g}(\mathbf{z}, \mathbf{z}^p | \mathbf{y}, \mathbf{y}^p) &= \prod_{i=1}^{d^p} (\mathcal{H}(-\mathbf{y}_i^p) f_i(\mathbf{z} | \mathbf{y}, -\mathbf{y}_i^p) + \mathcal{H}(\mathbf{y}_i^p) g_i(\mathbf{z} | \mathbf{y}, \mathbf{y}_i^p)),
 \end{aligned} \tag{29}$$

where  $f_i(\mathbf{z} | \mathbf{y}, \mathbf{y}_i^p)$  and  $g_i(\mathbf{z} | \mathbf{y}, \mathbf{y}_i^p)$  are conditional probability distributions defined by the value of  $\mathbf{y}_i^p$ .

This definition of  $\tilde{f}$  and  $\tilde{g}$  is a direct generalization of the definition of (26) in the one dimensional case. Fitting this definition into the transition probability  $p(\mathbf{z}, \mathbf{z}^p | \mathbf{y}, \mathbf{y}^p; \Delta t)$  in (27), the generalized update rule is defined and described in Algorithm 4. Again, we have the anti-symmetric kernel  $A(\mathbf{y}, \mathbf{y}^p, \mathbf{z}, \mathbf{z}^p)$  as in (28). As we prove in Appendix C.2, this construction has  $\int_{\mathbb{R}^{d+d^p}} A(\mathbf{y}, \mathbf{y}^p, \mathbf{z}, \mathbf{z}^p) d\mathbf{y} d\mathbf{y}^p = 0$  even with our  $d^p$ -dimensional *continuous* auxiliary variables.

In summary, we can use (27) to devise a practical algorithm for sampling (Algorithm 4). In particular, if we define  $f_i(\mathbf{z} | \mathbf{y}, \mathbf{y}_i^p)$  and  $g_i(\mathbf{z} | \mathbf{y}, \mathbf{y}_i^p)$  that are easy to sample from, then we can use the definitions of  $\tilde{f}$  and  $\tilde{g}$  in (29) to propose samples in the same way as the MH algorithm. After multiple rejections in  $\mathbf{y}$ , we resample  $\mathbf{y}^p$  according to  $\pi(\mathbf{y}^p)$  for a faster-mixing Markov chain in  $\mathbf{y}$ .

In multiple dimensions, a favorable direction of exploration is often not clear. Hence we suggest to take  $d^p = d$  as used in our experiment, so that  $\mathbf{z}^p$  has the same dimension as  $\mathbf{z}$ . Thus all directions can be explored by resampling the auxiliary variable  $\mathbf{z}^p$  after multiple rejections. This setting also helps to avoid the possibility of the resulting Markov chain being reducible. Also, when  $d^p = d$ ,  $f_i$  and  $g_i$  can be designed as:  $f_i(\mathbf{z} | \mathbf{y}) = f_i(\mathbf{z}_i | \mathbf{y}_i)$ , and  $g_i(\mathbf{z} | \mathbf{y}) = g_i(\mathbf{z}_i | \mathbf{y}_i)$ , depending only on  $\mathbf{z}_i$  and  $\mathbf{y}_i$ . Sample values in each dimension can thus be independently generated according to  $f_i(\mathbf{z}_i | \mathbf{y}_i)$  or  $g_i(\mathbf{z}_i | \mathbf{y}_i)$ . When a favorable direction of exploration *can* be determined (e.g., in the irreversible MALA algorithm in Section 5.2), we can take  $d^p = 1$ . Then  $\mathbf{z}^p$  belongs to a binary set  $\{-1, 1\}$ , rendering Algorithm 4 the same as the simpler version, Algorithm 3, which is the continuous state space generalization of the *lifting* method [Turitsyn et al., 2011, Vucelja, 2015].

In the experiments of Section 7, we take  $d^p = d$ ,  $f_i(\mathbf{z}(\ast) | \mathbf{z}(t), \mathbf{z}^p(t))$  as  $(\mathbf{z}_i(\ast) - \mathbf{z}_i(t)) / \mathbf{z}_i^p(t) \sim \Gamma(\alpha, \beta)$ ;  $g_i(\mathbf{z}(\ast) | \mathbf{z}(t), \mathbf{z}^p(t))$  as  $(\mathbf{z}_i(t) - \mathbf{z}_i(\ast)) / \mathbf{z}_i^p(t) \sim \Gamma(\alpha, \beta)$  and let  $\pi(\mathbf{z}^p)$  to be a restricted uniform distribution on the set  $\left\{ \mathbf{z}^p \left| \frac{1}{N} \|\mathbf{z}^p\|_1 = 1 \right. \right\}$ . Here  $\tilde{f}$  and  $\tilde{g}$  are designed to have no overlap in their support, maximizing the irreversibility effect. The norm of  $\mathbf{z}^p$  is set to be constant to ensure that  $\mathbf{z}^p$  contributes to the exploration of direction, instead of the expected distance of jump. It is worth noting that the accept-reject step in the current setting is the same as in random-walk MH.

## 5 Acceptance-Rejection Algorithms with General SDE proposals

As discussed in Section 1, there are various ways to combine the continuous dynamics with jump processes to propose new samplers. Since the Markov processes constructed in Section 3 and 4 can all be non-autonomous (resulting in time dependent matrices  $D$ ,  $Q$  and kernel functions  $S$ ,  $A$ ) as long as the stationary processes converge to the target distribution, one can iteratively follow continuous dynamics and jump processes to propose samples. With ergodicity, averages with respect to the sample values converge to averages with respect to the distribution. This is what is done in HMC: The complete dynamics include (i) a continuous Hamiltonian system with  $Q$  equal to the symplectic matrix, and (ii) a jump process in the auxiliary variable  $r$  with the symmetric kernel  $S(r_y, r_z) = \pi(r_y)\pi(r_z)/\Delta t$ ; the latter corresponds to the resampling of  $r$ . Alternating between the two processes provides the HMC method with exploration across the target distribution and ergodicity. (Note that an important consequence of the momentum reversal and resampling, however, is that the resulting HMC dynamics are *reversible*. In contrast, alternating between our proposed continuous and jump processes can still lead to irreversible dynamics.) Another straightforward way of combining our continuous and jump processes is to use the continuous dynamic sampler for some variables (e.g., real-valued variables) and the jump process sampler for others (e.g., discrete-valued variables).

In addition to the aforementioned means of combining continuous dynamics with jump processes for sampling, in this section we discuss how to use the continuous dynamics as a proposal distribution in our irreversible jump process accept-reject scheme of Algorithm 3, even when the continuous dynamics are not reversible. Previously, similar methods, such as the Metropolis-adjusted Langevin diffusion (MALA) and Riemannian Metropolis-adjusted Langevin diffusion (RMALA) [Roberts and Stramer, 2002, Xifara et al., 2014, Girolami and Calderhead, 2011], have only been proposed for reversible processes. These methods use one step integration of reversible SDEs to propose samples within a MH algorithm that accepts or rejects the proposal. In this section, we extend these methods to include proposals from any SDE in the form of (12) (any SDE with a mild integrability condition), without the requirement of reversibility. In Section 7, we show that this combination can generate better results in terms of rapid and efficient exploration of a distribution.

### 5.1 General SDE Proposals under Small Step Size Limit

Our ultimate goal is to use the stochastic dynamics of (12) to propose samples in the framework of Algorithm 3. In practice, we need to simulate from the discretized SDE of (15). Before analyzing this case, we first examine what would happen if we could *exactly* simulate the SDE of (12).

Here, we take  $f(\mathbf{z}|\mathbf{y}, \mathbf{y}^p)$  in (29) to be a Markov transition kernel  $P(\mathbf{z}|\mathbf{y}; dt)$  defined via an infinitesimal step  $dt$  in the SDE:

$$d\mathbf{z} = \left[ - (D(\mathbf{z}) + Q(\mathbf{z}))\nabla H(\mathbf{z}) + \Gamma(\mathbf{z}) \right] dt + \sqrt{2D(\mathbf{z})}d\mathbf{W}(t), \quad (30)$$

where  $\Gamma_i(\mathbf{z}) = \sum_j \frac{\partial}{\partial \mathbf{z}_j} (D_{ij}(\mathbf{z}) + Q_{ij}(\mathbf{z}))$ .

For the reverse proposal  $g(\mathbf{z}|\mathbf{y}, \mathbf{y}^p)$  in (29), we use the adjoint process  $P^\dagger(\mathbf{z}|\mathbf{y}; dt)$ , inverting the irreversible dynamics via  $Q(\mathbf{z}) \rightarrow -Q(\mathbf{z})$  [Ma and Qian, 2015]:

$$d\mathbf{z} = \left[ - (D(\mathbf{z}) - Q(\mathbf{z}))\nabla H(\mathbf{z}) + \tilde{\Gamma}(\mathbf{z}) \right] dt + \sqrt{2D(\mathbf{z})}d\mathbf{W}(t), \quad (31)$$

where  $\tilde{\Gamma}_i(\mathbf{z}) = \sum_j \frac{\partial}{\partial \mathbf{z}_j} (D_{ij}(\mathbf{z}) - Q_{ij}(\mathbf{z}))$ .



**Theorem 2.** For the Markov processes  $P(\mathbf{z}^{(T)}|\mathbf{z}^{(t)}; (T-t))$  and  $P^\dagger(\mathbf{z}^{(T)}|\mathbf{z}^{(t)}; (T-t))$  defined by the SDEs of (30) and (31) through Itô integral, the following equality holds:

$$\frac{P(\mathbf{z}^{(T)}|\mathbf{z}^{(t)}; (T-t))}{P^\dagger(\mathbf{z}^{(t)}|\mathbf{z}^{(T)}; (T-t))} = \frac{\pi(\mathbf{z}^{(T)})}{\pi(\mathbf{z}^{(t)})}. \quad (32)$$

The proof is in Appendix D. Using Theorem 2, we have

$$\alpha(\mathbf{z}^{(t)}, \mathbf{z}^*) = \min \left\{ 1, \frac{\pi(\mathbf{z}^*) P^\dagger(\mathbf{z}^{(t)}|\mathbf{z}^*; (T-t))}{\pi(\mathbf{z}^{(t)}) P(\mathbf{z}^*|\mathbf{z}^{(t)}; (T-t))} \right\} = 1.$$

Even though in Section 3 we saw that SDEs of the form in (12) have  $\pi(\mathbf{z})$  as the invariant distribution, it is not immediately obvious that using this SDE as a proposal in Algorithm 3 would lead to an acceptance rate of 1. This is a nice result, however, because it gives us insight into the fact that using more accurate numerical integrators could lead to higher acceptance rates. In Section 5.2, we analyze the accept-reject scheme for the simple first-order integration of (15) with finite step size  $\Delta t$ .

## 5.2 Generalizing the Metropolis-Adjusted Langevin Algorithm to Irreversible MALA

Since in practice we rely on finite step sizes  $\Delta t > 0$ , there will be numerical error and  $\frac{P(\mathbf{z}^*|\mathbf{z}^{(t)}; \Delta t)}{P^\dagger(\mathbf{z}^{(t)}|\mathbf{z}^*; \Delta t)}$  can differ from  $\frac{\pi(\mathbf{z}^*)}{\pi(\mathbf{z}^{(t)})}$ . We now propose a generalization of the MALA algorithm to correct for these errors. We make use of Algorithm 3 and take a general SDE and its adjoint process defined in Section 5.1 to propose samples using a one-step numerical integration (as in MALA). Because we have the local gradient information in the SDEs to guide us, the direction of the exploration is determined. So, we simply use a 1-dimensional discrete auxiliary variable  $\mathbf{y}^p$ , and thus the use of Algorithm 3 instead of the more general Algorithm 4. We call the resulting algorithm the *irreversible MALA* method.

Assuming a one-step numerical integration uses a  $\Delta t$  period of time, then the discretization of the SDE of (30) leads to

$$P(\mathbf{z}|\mathbf{y}; \Delta t) \propto \exp \left( -\frac{1}{4\Delta t} \mathbf{G}(\mathbf{y}, \mathbf{z})^T D(\mathbf{y})^{-1} \mathbf{G}(\mathbf{y}, \mathbf{z}) \right), \quad (33)$$

where

$$\mathbf{G}(\mathbf{y}, \mathbf{z}) = (\mathbf{z} - \mathbf{y}) - \left[ - (D(\mathbf{y}) + Q(\mathbf{y})) \nabla H(\mathbf{y}) + \Gamma(\mathbf{y}) \right] \Delta t.$$

Importantly, this allows us to compute  $f(\mathbf{z}^*|\mathbf{z}^{(t)}) = P(\mathbf{z}^*|\mathbf{z}^{(t)}; \Delta t)$  in Algorithm 3. The corresponding calculation for the adjoint process with the SDE in (31) is:

$$P^\dagger(\mathbf{z}|\mathbf{y}; \Delta t) \propto \exp \left( -\frac{1}{4\Delta t} \mathbf{G}^\dagger(\mathbf{y}, \mathbf{z})^T D(\mathbf{y})^{-1} \mathbf{G}^\dagger(\mathbf{y}, \mathbf{z}) \right), \quad (34)$$

where

$$\mathbf{G}^\dagger(\mathbf{y}, \mathbf{z}) = (\mathbf{z} - \mathbf{y}) - \left[ - (D(\mathbf{y}) - Q(\mathbf{y})) \nabla H(\mathbf{y}) + \Gamma(\mathbf{y}) \right] \Delta t.$$

This allows us to compute  $g(\mathbf{z}^*|\mathbf{z}^{(t)}) = P^\dagger(\mathbf{z}^*|\mathbf{z}^{(t)}; \Delta t)$ . The resulting irreversible MALA algorithm is summarized in Algorithm 5.

---

**Algorithm 5:** Irreversible MALA

---

```
randomly pick  $z^p$  from  $\{1, -1\}$  with equal probability
for  $t = 0, 1, 2 \dots N_{iter}$  do
    optionally, periodically resample auxiliary variable  $z^p \sim \mathcal{U}_{\{1, -1\}}$ 
    sample  $u \sim \mathcal{U}_{[0, 1]}$ 
    if  $z^p > 0$  then
        sample  $\eta_t \sim \mathcal{N}(0, 2\epsilon_t D(\mathbf{z}_t))$ 
         $\mathbf{z}(\ast) \leftarrow \mathbf{z}_t - \epsilon_t [(D(\mathbf{z}_t) + Q(\mathbf{z}_t)) \nabla H(\mathbf{z}_t) + \Gamma(\mathbf{z}_t)] + \eta_t$ 
         $\alpha(\mathbf{z}(t), \mathbf{z}(\ast)) = \min \left\{ 1, \frac{\pi(\mathbf{z}(\ast)) P^\dagger(\mathbf{z}(t) | \mathbf{z}(\ast); \Delta t)}{\pi(\mathbf{z}(t)) P(\mathbf{z}(\ast) | \mathbf{z}(t); \Delta t)} \right\}$ 
    end
    else
        sample  $\eta_t \sim \mathcal{N}(0, 2\epsilon_t D(\mathbf{z}_t))$ 
         $\mathbf{z}(\ast) \leftarrow \mathbf{z}_t - \epsilon_t [(D(\mathbf{z}_t) - Q(\mathbf{z}_t)) \nabla H(\mathbf{z}_t) + \Gamma(\mathbf{z}_t)] + \eta_t$ 
         $\alpha(\mathbf{z}(t), \mathbf{z}(\ast)) = \min \left\{ 1, \frac{\pi(\mathbf{z}(\ast)) P(\mathbf{z}(t) | \mathbf{z}(\ast); \Delta t)}{\pi(\mathbf{z}(t)) P^\dagger(\mathbf{z}(\ast) | \mathbf{z}(t); \Delta t)} \right\}$ 
    end
    if  $u < \alpha(\mathbf{z}(t), \mathbf{z}(\ast))$ ,  $\mathbf{z}(t+1) = \mathbf{z}(\ast)$ ;  $z^p(t+1) = z^p(t)$ 
    else  $\mathbf{z}(t+1) = \mathbf{z}(t)$ ;  $z^p(t+1) = -z^p(t)$ 
end
```

---

We know from Section 5.1 that in the small  $\Delta t$  limit,  $\alpha(\mathbf{z}^{(t)}, \mathbf{z}^\ast) = 1$ . Indeed,

$$\begin{aligned} & \frac{P(\mathbf{z} | \mathbf{y}; \Delta t)}{P^\dagger(\mathbf{y} | \mathbf{z}; \Delta t)} \cdot \frac{\pi(\mathbf{y})}{\pi(\mathbf{z})} \\ &= \exp \left( \frac{1}{4\Delta t} (\mathbf{G}^\dagger(\mathbf{z}, \mathbf{y})^T D(\mathbf{z})^{-1} \mathbf{G}^\dagger(\mathbf{z}, \mathbf{y}) - \mathbf{G}(\mathbf{y}, \mathbf{z})^T D(\mathbf{y})^{-1} \mathbf{G}(\mathbf{y}, \mathbf{z})) \right) \frac{\pi(\mathbf{y})}{\pi(\mathbf{z})} \rightarrow 1. \end{aligned}$$

From this, we see that there seems to be a step-size/acceptance-rate tradeoff. As mentioned in Section 5.1, a higher-order numerical scheme could potentially increase the acceptance rate with the same step size (see e.g. [Durmus et al., 2016] for optimal scaling of the MALA algorithm). We leave this as a direction for future research.

## 6 Related Work

There have been previous efforts to construct irreversible Markov chains for sampling. One example is using continuous dynamics to achieve this goal, which has been studied extensively. One can make use of Hamiltonian or generalized Hamiltonian dynamics to introduce irreversibility into the sampling procedure [Hwang et al., 1993, Hwang et al., 2005, Rey-Bellet and Spiliopoulos, 2015, Duncan et al., 2016, Ottobre et al., 2016] (Note that, as mentioned in Section 5, HMC results in reversible dynamics). There have been other samplers that utilize irreversible continuous dynamics such as underdamped Langevin [Horowitz, 1991, Chen et al., 2014] and Nosé-Hoover [Ding et al., 2014, Shang et al., 2015] dynamics, although irreversibility was not the emphasis in these works. As described in Section 3, any dynamic process that preserves the target distribution as the invariant measure can be used to devise an irreversible sampler within our framework.

We have also discussed using jump processes for sampling tasks. However, only recently have researchers constructed irreversible jump processes that form valid sampling procedures. In the *non-reversible MH* algorithm

[Bierkens, 2015], a vorticity function (or matrix) is added to the MH procedure. Hence, the difficulty of construction is translated to defining a valid vorticity function, similar to the difficulty of defining the antisymmetric kernel  $A(\mathbf{y}, \mathbf{z})$ . For the multivariate Gaussian distribution, the author discretized an irreversible Ornstein-Uhlenbeck process to obtain a suitable vorticity function. The *lifting method* [Turitsyn et al., 2011, Vucelja, 2015] makes a replica of the original state space ( $\mathbb{R}^d \times \{-1, 1\}$ ) to facilitate irreversibility in the sampling procedure. A skew detailed balance condition is imposed to ensure a valid antisymmetric kernel  $A(\mathbf{y}, \mathbf{z})$  in the expanded state space. The authors showed an example of applying the method to spin models. For both the *non-reversible MH* and *lifting* methods, it is unclear how to come up with a practical, easy-to-construct algorithm to handle a broad set of target distributions. In contrast, our irreversible jump sampler combines the idea of augmenting the state space (to  $\mathbb{R}^d \times \mathbb{R}^{d^p}$ ) with an accept-reject procedure similar to the MH algorithm to create an algorithm that is straightforward to implement for general target distributions.

Recently, the combined approach of using both continuous dynamics and jump processes has been proposed for constructing irreversible samplers. The *bouncy particle* [Bouchard-Côté et al., 2016] and *Zig-Zag* [Bierkens and Roberts, 2016, Bierkens et al., 2016] samplers use deterministic dynamics (irreversible in nature) combined with a Poisson process to create valid MCMC procedures. These two methods alternate between continuous dynamics and a Poisson jump process with an inhomogeneous rate (or intensity) to ensure the invariance of the target distribution. Our irreversible MALA algorithm avoids the difficulty of sampling from a Poisson process. Additionally, we end up with an algorithm that is a simple modification of vanilla MH, making it straightforward to use and plug in to existing algorithmic frameworks.

## 7 Experiments

To examine the correctness and attributes of our irreversible jump sampler (Algorithm 4), we consider various simulated scenarios, including the challenging cases of heavy tailed, multimodal, and correlated distributions. As mentioned in Section 4.3, we take  $f_i(\mathbf{z}(*)|\mathbf{z}(t), \mathbf{z}^p(t))$  as  $(\mathbf{z}_i(*) - \mathbf{z}_i(t))/\mathbf{z}_i^p(t) \sim \Gamma(\alpha, \beta)$ ;  $g_i(\mathbf{z}(*)|\mathbf{z}(t), \mathbf{z}^p(t))$  as  $(\mathbf{z}_i(t) - \mathbf{z}_i(*))/\mathbf{z}_i^p(t) \sim \Gamma(\alpha, \beta)$  and let  $\pi(\mathbf{z}^p)$  to be a restricted uniform distribution on the set  $\left\{ \mathbf{z}^p \left| \frac{1}{N} \|\mathbf{z}^p\|_1 = 1 \right. \right\}$ .

In other words, we choose a direction according to the auxiliary variable  $\mathbf{z}_i^p$  and propose gamma increments. If a proposal is rejected, that direction is reversed. when using  $f_i$  as the proposal distribution, states  $\mathbf{z}_i$  are updated as:  $\mathbf{z}_i(*) = \mathbf{z}_i(t) + r\mathbf{z}_i^p(t)$ ; when using  $g_i$  as the proposal distribution, states  $\mathbf{z}_i$  are updated as:  $\mathbf{z}_i(*) = \mathbf{z}_i(t) - r\mathbf{z}_i^p(t)$ , where  $r \sim \Gamma(\alpha, \beta)$ .

The hyperparameters  $\alpha$  and  $\beta$  are chosen using a generic procedure without fine-tuning to the target. We take the gamma shape parameter to be  $\alpha = 1.1$ , and change the rate parameter  $\beta$  approximately as  $\beta \propto \sqrt{V}$  ( $V$  is the volume of the region we would like to explore). See Appendix E for further discussion.

In this section, we also explore the irreversible MALA algorithm, and focus this analysis to a correlated distribution case where there is complex geometry. See Section 7.4.

### 7.1 Visual Comparison of Samplers

We first perform a qualitative comparison between the MH algorithm, our irreversible jump sampler (with gamma proposals), and the irreversible MALA algorithm to provide insights into their differences. It is demonstrated in Fig. 2 that the standard MH sampler jumps around randomly, but does so within a local region of the previous sample and irrespective of previous (directions of) jumps, leading to slow exploration of the distribution. In contrast, our irreversible counterpart (using gamma proposals) more rapidly traverses the distribution by following the direction of the previous jump, until being rejected. Finally, the irreversible MALA algorithm provides an even smoother trajectory by using continuous dynamics in place of independent gamma proposals.

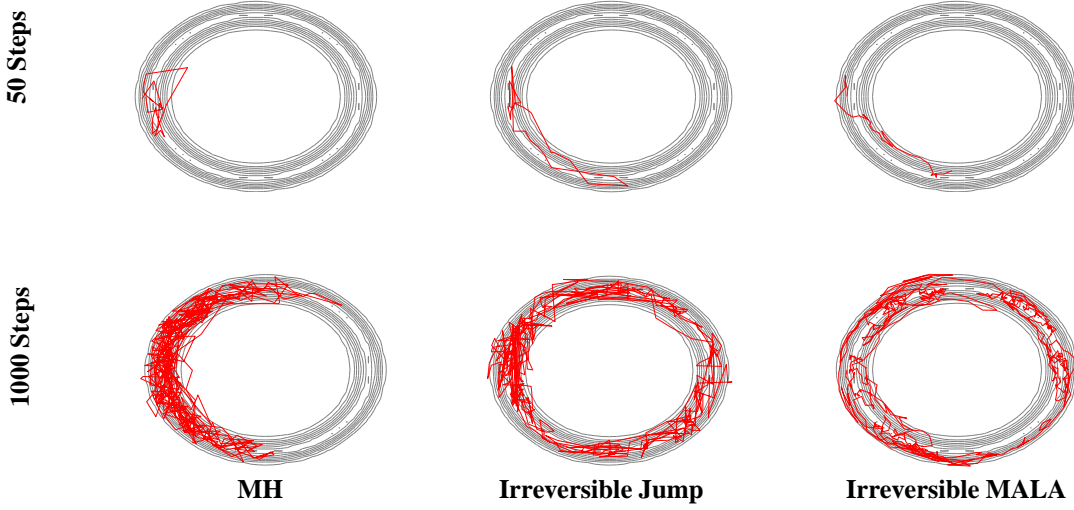


Figure 2: *Top row*: Trajectory of first 50 steps of (left) MH algorithm using Gaussian random walk proposals, (middle) irreversible jump algorithm with gamma proposals and (right) irreversible MALA algorithm. *Bottom row*: Similarly for the first 1000 steps of the algorithms.

Having visually examined the differences between the samplers to gain intuition, in what follows we provide a more quantitative analysis in the case of heavy-tailed, multimodal, and correlated distributions.

## 7.2 1D Heavy-tailed Distribution

We start by considering the task of sampling from 1D normal and log-normal distributions, the latter of which is a heavy-tailed distribution. The motivation for considering the simple 1D normal distribution is to validate the correctness of the sampler and to serve as a comparison relative to the heavy-tailed setting. We compare performance to a MH algorithm with normal proposals centered at the previous state. The results are shown in Fig. 3. Some may argue that the main possible benefit of our sampler arises from the gamma proposal distribution. To test this idea, we also compare against an MH algorithm using a symmetrized gamma proposal distribution:

$$(\mathbf{z}(\ast) - \mathbf{z}(t)) \sim \frac{1}{2} (f(\mathbf{z}(\ast)|\mathbf{z}(t)) + g(\mathbf{z}(\ast)|\mathbf{z}(t))).$$

We found that the irreversible jump sampler with the gamma proposals has better performance. In particular, the method can decrease autocorrelation without increasing the rejection rate (the rejection rate of all three methods are similar). The MH algorithm with symmetrized gamma proposals, on the other hand, leads to even higher autocorrelation than the vanilla MH algorithm. Intuitively, this result can be understood from Fig. 1: the irreversible algorithm leads to further exploration in one direction before circling back. (Also, see Fig. 2.)

For the heavy-tailed distribution, similar behavior is observed: the irreversible jump sampler converges to the desired distribution faster because its samples decorrelate more rapidly as a function of run time.

## 7.3 Multimodal Distributions

### 7.3.1 2D Bimodal distributions

We use our irreversible jump sampler to sample increasingly challenging bimodal distributions in 2D,  $\pi(z_1, z_2) = 2(z_1^2 - \tau)^2 - 0.2z_1 - 5z_1^2 + 5z_2^2$ , displayed in Fig. 4. Based on the results of Section 7.2, we simply compare against MH with random walk normal proposals and drop the symmetrized gamma proposal case. In Fig. 4 we see that the irreversible jump sampler significantly outperforms the random walk MH algorithm. Intuitively, this is facilitated by the greater traversing ability of the irreversible sampler, so that with the same acceptance rate, the irreversible

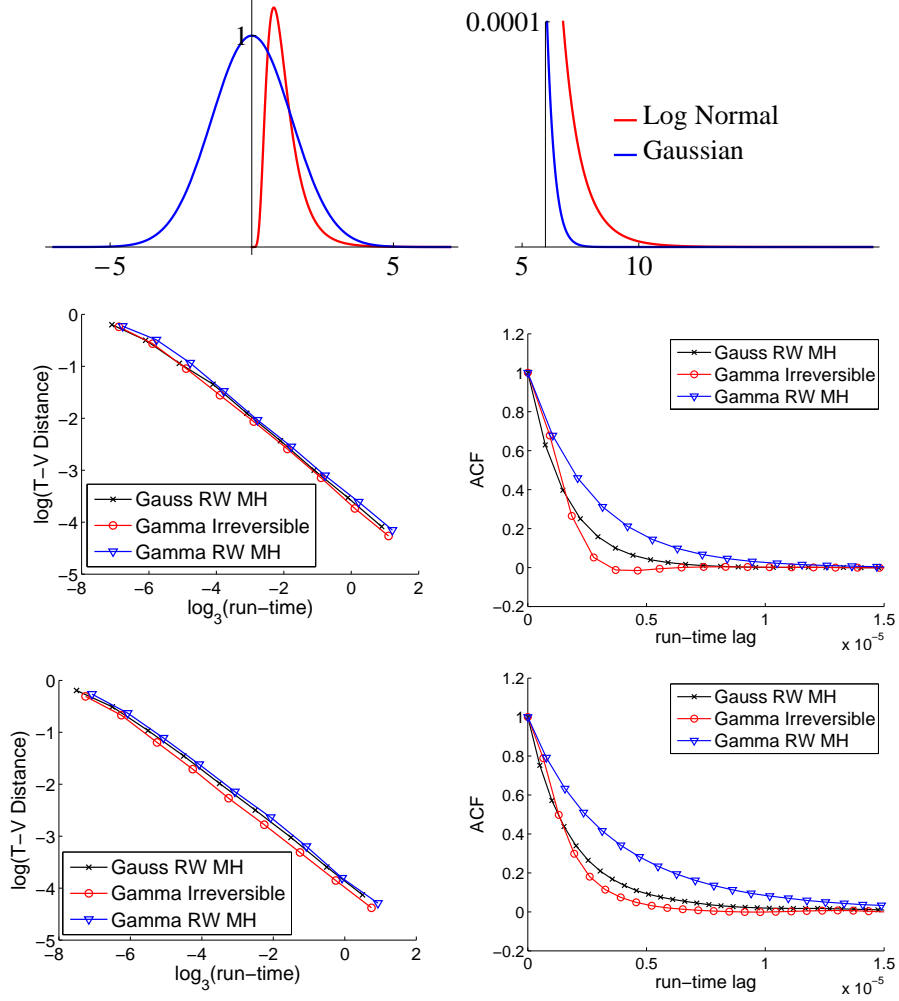


Figure 3: *Top row*: (Left) Normal and log-normal target distributions, and (right) zoom in of the tail distributions. *Middle row*: Results for normal target in terms of log total variation distance (T-V distance) vs. log run time (left) and ACF vs. lag in run time (right). *Bottom row*: Analogous plots for log normal target. Comparisons are made among the irreversible jump sampler of Algorithm 4 (Gamma Irreversible), random walk MH algorithm with Gaussian proposals (Gauss RW MH), and random walk MH algorithm with symmetrized gamma proposals (Gamma RW MH). Run time is measured in seconds.

sampler can explore more possible states than the reversible sampler, and have greater chance of transiting into another mode.

One way to capture this difference in the bimodal case is in terms of *escape time* from local modes, which we summarize in Table 1. We see that the irreversible jump sampler has escape times orders of magnitude lower. Furthermore, these escape times increase at a much smaller rate as the local modes become more concentrated, indicating much more rapid mixing between modes.

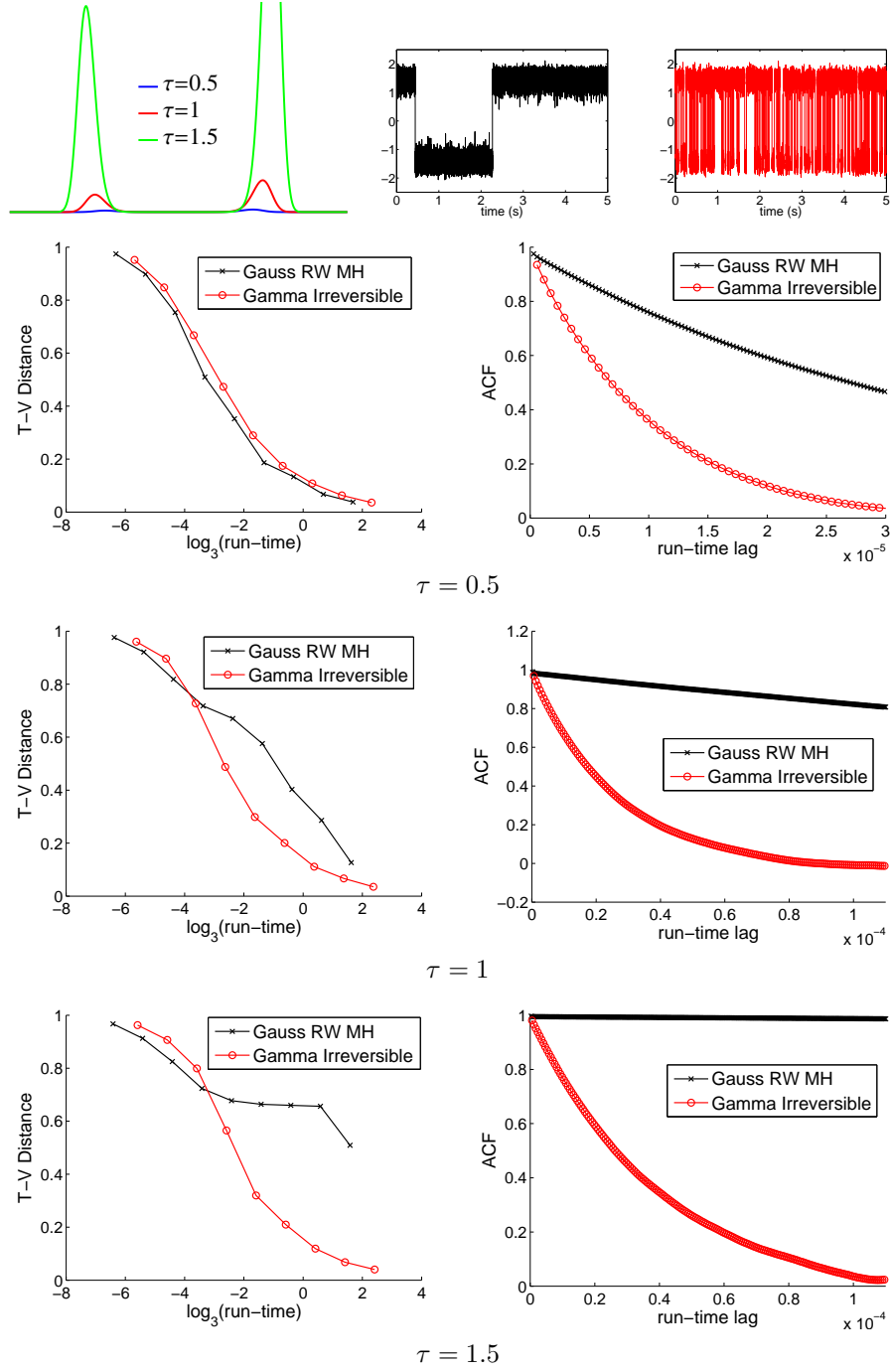


Figure 4: *Top row:* (Left) Bimodal targets,  $\pi(z_1, z_2) = 2(z_1^2 - \tau)^2 - 0.2z_1 - 5z_1^2 + 5z_2^2$ , for various values of  $\tau$ . Here we demonstrate a 1D cross section of the 2D distribution. (Middle) Sample state trajectories for MH and (right) irreversible jump sampler for the  $\tau = 1$  case. *Bottom rows:* Total variational distance vs. log run time (left) and ACF vs. lag in run time (right), with each row corresponding to a specific choice of  $\tau$ .

$\tau$	Avg. Escape Time for Irr. Sampler	Avg. Escape Time for MH Sampler
0.5	$1.94 \times 10^2$	$1.06 \times 10^3$
1	$4.64 \times 10^2$	$2.47 \times 10^4$
1.5	$9.06 \times 10^2$	$7.89 \times 10^5$
2	$2.41 \times 10^3$	N/A

Table 1: Comparison of average escape time from one local mode to another between the irreversible jump sampler and random walk MH. The distribution in 2D is more challenging with bigger values of  $\tau$  (plotted in Fig. 4). “N/A” in the last entry means that the escape time is so long that an accurate estimate of it is not available.

### 7.3.2 2D Multimodal distributions

We also tested our method against a recently considered multimodal setting [Tak et al., 2016]. In the first setting considered, the target distribution is highly multimodal in 2D with unevenly distributed modes. Furthermore, the high mass modes have smaller radii of variation. In the second setting considered, these modes are highly concentrated and well separated, which is an extremely challenging setting for most samplers. See Figs. 5 and 6. In [Tak et al., 2016], a *repulsive-attractive Metropolis (RAM)* sampler was proposed with a structure specifically designed to efficiently handle these types of multimodal distributions. We use this as a gold-standard comparison, since this method was already shown to outperform parallel tempering and alternatives [Kou et al., 2006] in this setting.

We focus our performance analysis on the decay speed of the autocorrelation function (ACF). This can be understood by taking the Gaussian random walk MH algorithm as an example: Although the Gaussian random walk MH algorithm seems to perform well in terms of convergence of total variation distance, this effect is based on exploring one mode really well in a short period of time, instead of making more distant moves to explore other modes. In contrast, the ACF better characterizes the exploration of the samples through the whole space.

Our results are summarized in Figs. 5 and 6 for each of the two simulated multimodal scenarios. In the first scenario, our sampler outperforms both MH and RAM. In the second scenario, where we have highly concentrated and separated modes, the RAM method tailored to this scenario slightly outperforms our approach. Overall, however, the irreversible jump sampler provides surprisingly good performance in these scenarios despite not having been designed specifically for this setting.

## 7.4 Correlated Distribution

We also test our algorithm on a highly correlated (moon-shaped) target distribution, where  $\pi(z_1, z_2) = z_1^4/10 + (4(z_2 + 1.2) - z_1^2)^2/2$ . In terms of number of iterations, the irreversible jump sampler with gamma proposals decorrelates and converges to the posterior distribution faster (c.f. Fig. 8 in Appendix E.2). However, in terms of run time, our sampler does not perform as well as random walk MH algorithm, as explored in Fig. 7. The reason is that the correlated distribution has complex geometry. Faster exploration in random directions, as provided by our irreversible sampler with independent proposals, only marginally increases the mixing effect in each step relative to the reversible independent proposals of MH. Since the calculation of the distribution is not demanding in this case, the small overhead of the irreversible sampler (keeping track of the number of rejections and resample the direction of exploration after multiple rejections) actually makes a difference and thus results in our sampler with gamma proposals providing slightly worse performance in terms of runtime.

To improve the performance of our irreversible sampler further in this correlated target case, it would be appealing to take the geometric information about the level sets—including the higher mass regions—into account. Indeed, we are able to do this by replacing the independent gamma proposals with proposals from our continuous dynamics sampler, as described in Section 5. To demonstrate the effect of irreversibility, we choose



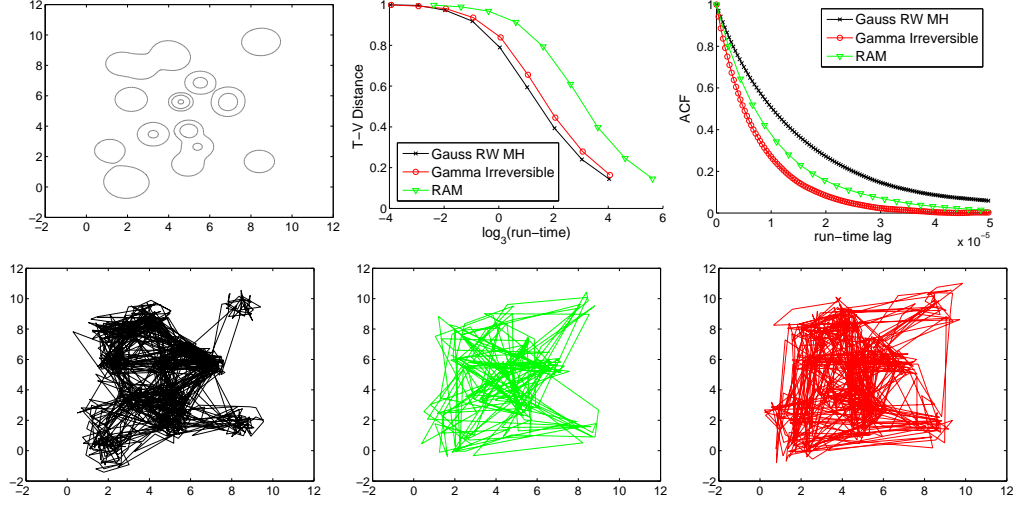


Figure 5: *Top row:* (Left) Contour plot of a challenging multimodal probability density function; (middle) T-V distance and ACF comparisons among Gauss RW MH algorithm, Gamma Irreversible, and the recently proposed repulsive-attractive Metropolis (RAM) sampler. *Bottom row:* A sample run of all three samplers, respectively.

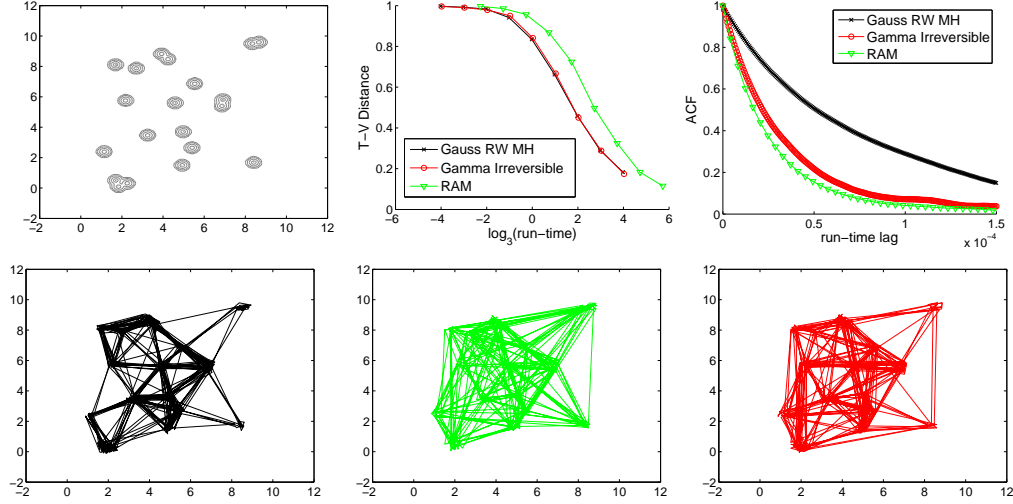


Figure 6: Plots as in Fig. 5, but for an even more challenging multimodal case where the modes are very concentrated and well separated.

$D(\mathbf{z}) = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$  and  $Q(\mathbf{z}) = \begin{pmatrix} 0 & -2 \\ 2 & 0 \end{pmatrix}$  in Eqs. (30), (31), (33), and (34). In this case, our irreversible MALA algorithm (Algorithm 5) significantly outperforms the Gaussian random walk MH, as well as HMC [Neal, 2010] and the standard reversible MALA algorithm [Roberts and Stramer, 2002]. Because the target distribution has complex geometry, the continuous dynamics can provide guidance on locating the higher mass regions and exploring the contours rapidly with the gradient information. HMC and MALA algorithms exploit this effect,

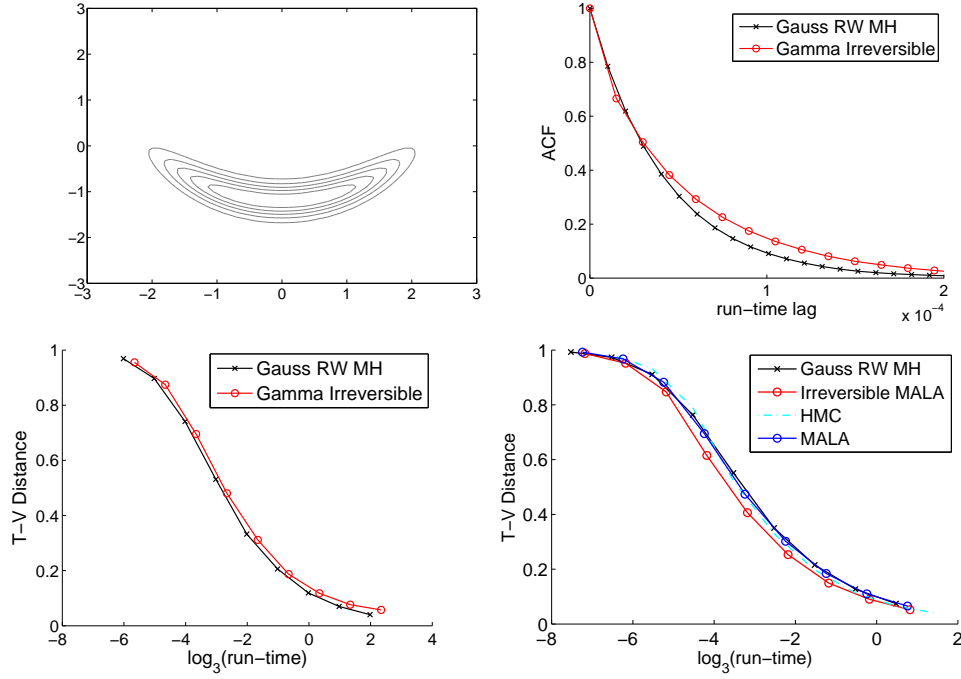


Figure 7: *Top row*: Correlated distribution with complex geometry in 2D,  $\pi(z_1, z_2) = z_1^4/10 + (4(z_2 + 1.2) - z_1^2)^2/2$  (left) and ACF vs. lag in run time of Gamma Irreversible algorithm against Gauss RW MH (right). *Bottom row*: T-V distance vs. log run time. Comparisons are made between Gauss RW MH and Gamma Irreversible (left), and Gauss RW MH, Irreversible MALA, HMC, and MALA (right).

but we additionally see gains from the irreversibility of the sampler.

This experiment demonstrates the gains that are possible by combining our continuous dynamics and jump process frameworks, beyond what either can provide individually.

## 8 Scaling Up the Sampling Algorithms for Large Datasets

We wish to scale up the previously discussed sampling algorithms to cases where our target distribution is a posterior distribution in a Bayesian model, and we are faced with a huge number of observations  $\mathcal{S}$ . In this case, the likelihood, or its gradient, can be computationally prohibitive to compute. In the cases of i.i.d. data, we write our target distribution as  $\pi(\theta) = p(\mathcal{S}|\theta) p(\theta) = \prod_{s \in \mathcal{S}} p(s|\theta) p(\theta)$ . For the samplers designed from continuous dynamics (Section 3), we can use *stochastic gradients*, in place of the full data gradient as elaborated in [Ma et al., 2015] and outlined below. For samplers using jump processes (Section 4), we discuss a generalization of the subsampling-within-MH ideas in [Korattikara et al., 2014, Bardenet et al., 2014, Bardenet et al., 2015].

**Stochastic Gradient Samplers** For samplers using continuous dynamics (12), the computationally intensive component in the update rule of (15) is the computation of  $\nabla H(\theta) = -\nabla \log \pi(\theta) = -\sum_{s \in \mathcal{S}} \nabla \log p(s|\theta) - \nabla \log p(\theta)$ . One idea for avoiding this per iteration cost is to use *stochastic gradients* instead [Robbins and Monro, 1951]. Here, a noisy gradient based on a data subsample or *minibatch*, is used as an unbiased estimator of the full data gradient. More formally, we examine *independently sampled* minibatches  $\tilde{\mathcal{S}} \subset \mathcal{S}$ . The corresponding log-posterior

for these data is

$$\tilde{H}(\theta) = -\frac{|\mathcal{S}|}{|\tilde{\mathcal{S}}|} \sum_{s \in \tilde{\mathcal{S}}} \log p(s|\theta) - \log p(\theta); \quad \tilde{\mathcal{S}} \subset \mathcal{S}. \quad (35)$$

The specific form of (35) implies that  $\tilde{H}(\theta)$  is an unbiased estimator of  $H(\theta)$ , thus  $\nabla \tilde{H}(\theta)$  is an unbiased estimator of  $\nabla H(\theta)$ . The key question in many of the existing stochastic gradient MCMC algorithms is whether the noise injected by the stochastic gradient  $\nabla \tilde{H}(\theta)$  adversely affects the stationary distribution of the modified dynamics. One way to analyze the impact of the stochastic gradient is to assume the central limit theorem holds:  $\nabla \tilde{H}(\theta) = \nabla H(\theta) + \mathcal{N}(0, V(\theta))$ . Simply plugging in  $\nabla \tilde{H}(\theta)$  in place of  $\nabla H(\theta)$  in (15) results in dynamics with an additional noise term  $(D(\theta_t) + Q(\theta_t))[\mathcal{N}(0, V(\theta_t))]^T$ . In our earlier work [Ma et al., 2015], we studied the influence of this noise and showed that one may counteract it by assuming an estimate  $\hat{B}_t$  of the noise variance and following:

$$\mathbf{z}_{t+1} \leftarrow \mathbf{z}_t - \epsilon_t \left[ (D(\mathbf{z}_t) + Q(\mathbf{z}_t)) \nabla \tilde{H}(\mathbf{z}_t) + \Gamma(\mathbf{z}_t) \right] + \mathcal{N}(0, \epsilon_t(2D(\mathbf{z}_t) - \epsilon_t \hat{B}_t)). \quad (36)$$

In the limit of  $\epsilon_t$  going to zero, the stationary distribution is preserved. For finite  $\epsilon_t$ , a bias exists. The same bias-speed tradeoff was used in past stochastic gradient sampling methods [Welling and Teh, 2011, Chen et al., 2014, Ding et al., 2014, Shang et al., 2015]. In [Ma et al., 2015], we also devise methods for defining new samplers using existing  $D$  and  $Q$  matrices as building blocks. As a specific example, we consider a Riemann version of SGHMC method [Chen et al., 2014], and demonstrated the gains over existing benchmarks on a large streaming Wikipedia analysis. For the irreversible (and reversible) MALA algorithms, if an accurate estimate of the stochastic gradient noise is available, the stochastic gradient method can be combined with the subsampling approach described below to provide scalable variants of the MALA algorithms.

**Subsampling of Irreversible Sampler from Jump Processes** For the irreversible jump sampler (Algorithm 4), we can directly generalize the subsampling idea for the MH algorithms [Korattikara et al., 2014] and its adaptive and proxy method variations [Bardenet et al., 2014, Bardenet et al., 2015]. In Algorithm 4, the computational bottleneck is at the step where we decide to accept or reject the proposal from  $\tilde{f}(\theta(*), \theta^p(*)|\theta(t), \theta^p(t))$ , since we need to calculate  $\pi(\theta(*))/\pi(\theta(t))$ , requiring evaluation of the entire likelihood.

The accept-reject step is then implemented by sampling a uniform random variable  $u \sim \mathcal{U}_{[0,1]}$  and accepting the proposal if and only if  $u < \alpha(\theta(t), \theta^p(t), \theta(*), \theta^p(*))$ . Following [Korattikara et al., 2014, Bardenet et al., 2014], we can rewrite this condition as:

$$\Lambda_{\mathcal{S}}(\theta(t), \theta(*)) > \Theta_{\mathcal{S}}(u, \theta(t), \theta(*), \theta^p(t), \theta^p(*)), \quad (37)$$

where

$$\begin{aligned} \Lambda_{\mathcal{S}}(\theta(t), \theta(*)) &= \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \log \left[ \frac{p(s|\theta(*))}{p(s|\theta(t))} \right], \\ \Theta_{\mathcal{S}}(u, \theta(t), \theta(*), \theta^p(t), \theta^p(*)) &= \frac{1}{|\mathcal{S}|} \log \left[ u \frac{\pi(\theta^p(t)) \tilde{f}(\theta(*), \theta^p(*)|\theta(t), \theta^p(t))}{\pi(\theta^p(*)) \tilde{g}(\theta(t), \theta^p(t)|\theta(*), \theta^p(*))} \right]. \end{aligned}$$

For the computationally intractable  $\Lambda_{\mathcal{S}}(\theta(t), \theta(*))$ , we can use a subset of data to approximate it via:

$$\Lambda_{\tilde{\mathcal{S}}}^*(\theta(t), \theta(*)) = \frac{1}{|\tilde{\mathcal{S}}|} \sum_{s \in \tilde{\mathcal{S}}} \log \left[ \frac{p(s|\theta(*))}{p(s|\theta(t))} \right] \approx \Lambda_{\mathcal{S}}(\theta(t), \theta(*)); \quad \tilde{\mathcal{S}} \subset \mathcal{S}.$$

Importantly,  $|\Lambda_S(\theta(t), \theta(*)) - \Lambda_S^*(\theta(t), \theta(*))|$  can be bounded probabilistically [Bardenet et al., 2014]. Hence, a speed-bias tradeoff can be quantified through the probabilistic bounds when we use  $\Lambda_S^*(\theta(t), \theta(*))$  instead of  $\Lambda_S(\theta(t), \theta(*))$ . In some cases, more data can be used to tighten or approximate the bound on  $|\Lambda_S(\theta(t), \theta(*)) - \Lambda_S^*(\theta(t), \theta(*))|$ , so that inequality (37) can always be verified. Then the above procedure still yields an exact sampler. In many cases, however, so much data has to be used that the computation gains of subsampling are negligible. As such, we typically view this scheme as one with quantifiable bias. See [Bardenet et al., 2015] for further discussions and developments.

Overall, due to the similarities between our irreversible jump sampler and the MH algorithm, many methods developed specifically for MH can be applied in our context, which is quite appealing. For example, as discussed above, we have directly applied the subsampling approach designed for scaling MH to our approach. We can also combine our irreversible jump sampler with the RAM algorithm to further improve exploration in the case of multimodal targets.

## 9 Conclusion

In this paper, we proposed frameworks for MCMC algorithms with both continuous dynamics and jump processes. We analyzed each of these components separately, and then showed how to combine them. For each component, we decomposed the dynamics into reversible and irreversible processes, and with a parameterization that was easy to specify while ensuring the correct stationary distribution.

First, we found that any continuous Markov process (with a mild integrability condition) can always be parameterized by its stationary distribution (i.e. the target distribution)  $\pi(\mathbf{z})$ , a positive semi-definite diffusion matrix  $D(\mathbf{z})$ , and a skew-symmetric curl matrix  $Q(\mathbf{z})$ . We analyzed the properties of the process in terms of  $D(\mathbf{z})$  and  $Q(\mathbf{z})$ . In the context of Bayesian analysis with large datasets, we further discussed scalable methods using stochastic gradients.

Second, we turned to jump processes and considered a parameterization in terms of a symmetric kernel function  $S(\mathbf{y}, \mathbf{z})$  and an anti-symmetric kernel function  $A(\mathbf{y}, \mathbf{z})$ . We showed that when  $\int_{\mathbf{y}} A(\mathbf{y}, \mathbf{z}) d\mathbf{y} = 0$ , the jump process has the target distribution  $\pi(\mathbf{z})$  as its stationary distribution. When  $A(\mathbf{y}, \mathbf{z}) \neq 0$ , the jump process is irreversible. Facilitated by the framework, we constructed a new class of irreversible sampling algorithms that can be implemented similarly to the MH algorithm while directly satisfying  $\int_{\mathbf{y}} A(\mathbf{y}, \mathbf{z}) d\mathbf{y} = 0$ . Our experiments demonstrate that our proposed irreversible jump sampler is more efficient than the traditional reversible ones across a broad range of target distributions. We further discussed how a scalable variant is possible using the same subsampling idea as proposed for MH samplers.

Finally, we developed a technique to combine the continuous and jump processes by using the continuous dynamics as a proposal in the irreversible jump sampler. The directional effect of the continuous dynamics can facilitate better exploration of the target distribution than a simple proposal distribution. Likewise, one can also think of this framework as enabling a large step size to be taken in the continuous dynamic simulations while correcting for discretization error. We demonstrated that such a sampler can outperform samplers with independent proposals, samplers with continuous dynamics alone, or reversible versions of the combined approach (i.e., MALA).

The proposed framework requires a few critical choices to be made. For the continuous dynamics, we must specify  $D(\mathbf{z})$  and  $Q(\mathbf{z})$ . For the jump processes, the specific algorithm we proposed requires selecting proposal distributions  $\tilde{f}(\mathbf{z}, \mathbf{z}^p | \mathbf{y}, \mathbf{y}^p)$  and  $\tilde{g}(\mathbf{z}, \mathbf{z}^p | \mathbf{y}, \mathbf{y}^p)$ , and specifying the domain of the auxiliary variables  $\mathbf{y}^p$ . Our experiments have simply demonstrated that for certain choices of these matrices and parameters, we can achieve state-of-the-art performance in a variety of sampling tasks. An important direction for future work is to devise methods to analyze and explore the choices of these algorithmic parameters. For example, in higher dimensions, tuning the hyperparameters so that the irreversible jump sampler explores all dimensions efficiently is an interesting topic for further discussion and quantification. Also, in the irreversible MALA algorithm, we only used constant  $D(\mathbf{z})$  and  $Q(\mathbf{z})$ , but using adaptive  $D(\mathbf{z})$  and  $Q(\mathbf{z})$  could potentially result in more efficient sampling algorithms.

[[Girolami and Calderhead, 2011](#), [Ma et al., 2015](#)]. Another area of research is to examine using a higher order integration scheme in our irreversible MALA algorithm.

## **Acknowledgments**

This work was supported in part by ONR Grant N00014-15-1-2380, NSF CAREER Award IIS-1350133, DARPA Grant FA9550-12-1-0406 negotiated by AFOSR, and the TerraSwarm Research Center sponsored by MARCO and DARPA. We also thank Samuel Livingstone, Paul Fearnhead and Hong Qian for helpful suggestions and discussions.

## A Background

### A.1 Existence Conditions Leading to the Differential Chapman-Kolmogorov Equation

A more useful form of the CK equation can be obtained by assuming three mild existence conditions for all  $\epsilon > 0$ :

1.  $\lim_{\Delta t \rightarrow 0} p(\mathbf{x}|\mathbf{z}; \Delta t)/\Delta t = W(\mathbf{x}|\mathbf{z})$  exists uniformly in  $\mathbf{x}$  and  $\mathbf{z}$  for  $|\mathbf{x} - \mathbf{z}| \geq \epsilon$ ;
2.  $\lim_{\Delta t \rightarrow 0} \frac{1}{\Delta t} \int_{|\mathbf{x}-\mathbf{z}| < \epsilon} d\mathbf{x}(\mathbf{x}_i - \mathbf{z}_i) p(\mathbf{x}|\mathbf{z}; \Delta t) = \mathbf{f}_i(\mathbf{z}) + O(\epsilon)$ ;
3.  $\lim_{\Delta t \rightarrow 0} \frac{1}{\Delta t} \int_{|\mathbf{x}-\mathbf{z}| < \epsilon} d\mathbf{x}(\mathbf{x}_i - \mathbf{z}_i)(\mathbf{x}_j - \mathbf{z}_j) p(\mathbf{x}|\mathbf{z}; \Delta t) = 2D_{ij}(\mathbf{z}) + O(\epsilon)$  uniformly in  $\mathbf{z}$  and  $\epsilon$ .

## B Proof of Theorem 1 (equivalence of operators $\mathcal{L}[\cdot]$ and $\widehat{\mathcal{L}}[\cdot]$ with stationary distribution $\pi(\mathbf{z})$ )

Since  $\pi(\mathbf{z})$  is the stationary distribution of  $\frac{\partial}{\partial t} p(\mathbf{z}|\mathbf{y}; t) = \widehat{\mathcal{L}}[\frac{p(\mathbf{z}|\mathbf{y}; t)}{\pi(\mathbf{z})}]$ , we have:  $\widehat{\mathcal{L}}[1] = 0$ . That is:

$$\sum_i \frac{\partial}{\partial \mathbf{z}_i} \left\{ \sum_j \frac{\partial}{\partial \mathbf{z}_j} (D_{ij}(\mathbf{z})\pi(\mathbf{z})) - \mathbf{f}_i\pi(\mathbf{z}) \right\} = 0. \quad (38)$$

Expanding operator  $\widehat{\mathcal{L}}[\cdot]$ , we find that:

$$\begin{aligned} \widehat{\mathcal{L}}[\varphi(\mathbf{z})] &= \sum_i \frac{\partial}{\partial \mathbf{z}_i} \left\{ \sum_j \frac{\partial}{\partial \mathbf{z}_j} (D_{ij}(\mathbf{z})\pi(\mathbf{z})) - \mathbf{f}_i\pi(\mathbf{z}) \right\} \varphi(\mathbf{z}) + \sum_i \left\{ \sum_j \frac{\partial}{\partial \mathbf{z}_j} (D_{ij}(\mathbf{z})\pi(\mathbf{z})) - \mathbf{f}_i\pi(\mathbf{z}) \right\} \frac{\partial \varphi(\mathbf{z})}{\partial \mathbf{z}_i} \\ &\quad + \sum_{i,j} \left\{ \frac{\partial}{\partial \mathbf{z}_i} (D_{ij}(\mathbf{z})\pi(\mathbf{z})) \right\} \frac{\partial \varphi(\mathbf{z})}{\partial \mathbf{z}_j} + \sum_{i,j} D_{ij}(\mathbf{z})\pi(\mathbf{z}) \frac{\partial^2 \varphi(\mathbf{z})}{\partial \mathbf{z}_i \partial \mathbf{z}_j}. \end{aligned}$$

Since  $\sum_i \frac{\partial}{\partial \mathbf{z}_i} \left\{ \sum_j \frac{\partial}{\partial \mathbf{z}_j} (D_{ij}(\mathbf{z})\pi(\mathbf{z})) - \mathbf{f}_i\pi(\mathbf{z}) \right\} = 0$ ,

$$\widehat{\mathcal{L}}[\varphi(\mathbf{z})] = \sum_i \left\{ 2 \sum_j \frac{\partial}{\partial \mathbf{z}_j} (D_{ij}(\mathbf{z})\pi(\mathbf{z})) - \mathbf{f}_i\pi(\mathbf{z}) \right\} \frac{\partial \varphi(\mathbf{z})}{\partial \mathbf{z}_i} + \sum_{i,j} D_{ij}(\mathbf{z})\pi(\mathbf{z}) \frac{\partial^2 \varphi(\mathbf{z})}{\partial \mathbf{z}_i \partial \mathbf{z}_j}. \quad (39)$$

Expanding operator  $\mathcal{L}[\cdot]$  leads to:

$$\mathcal{L}[\varphi(\mathbf{z})] = \sum_{ij} \frac{\partial}{\partial \mathbf{z}_i} (D_{ij}(\mathbf{z}) + Q_{ij}(\mathbf{z})) \pi(\mathbf{z}) \frac{\partial \varphi(\mathbf{z})}{\partial \mathbf{z}_j} + \sum_{ij} (D_{ij}(\mathbf{z}) + Q_{ij}(\mathbf{z})) \pi(\mathbf{z}) \frac{\partial^2 \varphi(\mathbf{z})}{\partial \mathbf{z}_i \partial \mathbf{z}_j}.$$

Noting that matrices  $D(\mathbf{z})$  is symmetric and  $Q(\mathbf{z})$  is skew-symmetric, we have:

$$\mathcal{L}[\varphi(\mathbf{z})] = \sum_{ij} \frac{\partial}{\partial \mathbf{z}_j} (D_{ij}(\mathbf{z}) - Q_{ij}(\mathbf{z})) \pi(\mathbf{z}) \frac{\partial \varphi(\mathbf{z})}{\partial \mathbf{z}_i} + \sum_{ij} D_{ij}(\mathbf{z})\pi(\mathbf{z}) \frac{\partial^2 \varphi(\mathbf{z})}{\partial \mathbf{z}_i \partial \mathbf{z}_j}. \quad (40)$$

Comparing (39) and (40), we find that  $\mathcal{L}[\cdot]$  and  $\hat{\mathcal{L}}[\cdot]$  are equivalent when the following condition is satisfied:

$$\sum_j \frac{\partial}{\partial \mathbf{z}_j} Q_{ij}(\mathbf{z})\pi(\mathbf{z}) = \mathbf{f}_i\pi(\mathbf{z}) - \sum_j \frac{\partial}{\partial \mathbf{z}_j} \left( D_{ij}(\mathbf{z})\pi(\mathbf{z}) \right). \quad (41)$$

The nice forms of Eqs. (41) and (38) imply that the questions can be transformed into a linear algebra problem once we apply a Fourier transform to them. Denote the Fourier transform of  $Q(\mathbf{z})\pi(\mathbf{z})$  as  $\hat{Q}(\mathbf{k})$  and the Fourier transform of  $\mathbf{f}_i(\mathbf{z})\pi(\mathbf{z}) - \sum_j \frac{\partial}{\partial \mathbf{z}_j} \left( D_{ij}(\mathbf{z})\pi(\mathbf{z}) \right)$  as  $\hat{F}_i(\mathbf{k})$ , where  $\mathbf{k} = (\mathbf{k}_1, \dots, \mathbf{k}_d)^T$  is the set of the spectral variables. That is:

$$\begin{aligned} \hat{Q}_{ij}(\mathbf{k}) &= \int_{\mathcal{D}} Q_{ij}(\mathbf{z})\pi(\mathbf{z})e^{-2\pi i \mathbf{k}^T \mathbf{z}} d\mathbf{z}; \\ \hat{F}_i(\mathbf{k}) &= \int_{\mathcal{D}} \left( \mathbf{f}_i(\mathbf{z})\pi(\mathbf{z}) - \sum_j \frac{\partial}{\partial \mathbf{z}_j} \left( D_{ij}(\mathbf{z})\pi(\mathbf{z}) \right) \right) e^{-2\pi i \mathbf{k}^T \mathbf{z}} d\mathbf{z}. \end{aligned}$$

Here,  $i$  denotes the unit imaginary number;  $\pi$  denotes the Archimedes' constant, whereas  $\pi(\mathbf{z})$  is the stationary (the desired) distribution. Then,  $\frac{\partial}{\partial \mathbf{z}_j} \left( Q_{ij}(\mathbf{z})\pi(\mathbf{z}) \right)$  is transformed to  $2\pi i \hat{Q}_{ij}\mathbf{k}_j$ , and (41) becomes the following equivalent form in Fourier space:

$$\begin{cases} 2\pi i \hat{Q}\mathbf{k} = \hat{F} \\ \mathbf{k}^T \hat{F} = 0. \end{cases}$$

Hence, it is clear that matrix  $\hat{Q}$  must be a skew-symmetric projection matrix from the span of  $\mathbf{k}$  to the span of  $\hat{F}$ , where  $\mathbf{k}$  and  $\hat{F}$  are always orthogonal to each other. We thereby construct  $\hat{Q}$  as a combination of two rank one projection matrices:

$$\hat{Q} = (2\pi i)^{-1} \frac{\hat{F}\mathbf{k}^T}{\mathbf{k}^T \mathbf{k}} - (2\pi i)^{-1} \frac{\mathbf{k}\hat{F}^T}{\mathbf{k}^T \mathbf{k}}.$$

We arrive at the final result that matrix  $Q(\mathbf{z})$  is equal to  $\pi(\mathbf{z})^{-1}$  times the inverse Fourier transform of  $\hat{Q}(\mathbf{k})$ :

$$Q_{ij}(\mathbf{z}) = \pi(\mathbf{z})^{-1} \int_{\mathcal{D}} \frac{\mathbf{k}_j \hat{F}_i(\mathbf{k}) - \mathbf{k}_i \hat{F}_j(\mathbf{k})}{(2\pi i) \cdot \sum_l \mathbf{k}_l^2} e^{2\pi i \sum_l \mathbf{k}_l \mathbf{z}_l} d\mathbf{k}. \quad (42)$$

Thus, if  $\left( \mathbf{f}_i(\mathbf{z})\pi(\mathbf{z}) - \sum_j \frac{\partial}{\partial \mathbf{z}_j} \left( D_{ij}(\mathbf{z})\pi(\mathbf{z}) \right) \right)$  belongs to the space of  $L^1$ , then operator  $\hat{\mathcal{L}}[\cdot]$  with any  $D(\mathbf{z})$  and  $\mathbf{f}(\mathbf{z})$  can be turned into the new formulation of  $\mathcal{L}[\cdot]$  with  $Q(\mathbf{z})$ .

**Remark 1.** Entries in the skew-symmetric projector  $Q_{ij}(\mathbf{z})$  constructed in (42) are real: Denote  $\mathbf{a}_i^2 = \sum_{l \neq i} \mathbf{k}_l^2$ , then

the inverse Fourier transform of  $\frac{\mathbf{k}_i}{(2\pi i) \cdot \sum_l \mathbf{k}_l^2}$  along the partial variable  $\mathbf{k}_i$  is equal to:

$$\mathbf{g}_i(\mathbf{z}) = -\frac{1}{2}e^{-2\pi \mathbf{a}_i \mathbf{z}_i} \mathcal{H}[\mathbf{z}_i] + \frac{1}{2}e^{2\pi \mathbf{a}_i \mathbf{z}_i} \mathcal{H}[-\mathbf{z}_i],$$

where  $\mathcal{H}[x]$  is the Heaviside function. Because  $\mathbf{g}_i(\mathbf{z})$  is an even function in  $k_l$ ,  $l \neq i$ , its total inverse Fourier transform is real. Therefore, the inverse Fourier transform of  $\frac{\mathbf{k}_i \hat{F}_j(\mathbf{k})}{(2\pi i) \cdot \sum_l \mathbf{k}_l^2}$  is the convolution of two real functions.



## C Irreversible Jump Processes for MCMC

### C.1 Equivalence of $\mathcal{J}[\cdot]$ and $\hat{\mathcal{J}}[\cdot]$

We decompose operator  $\hat{\mathcal{J}}[\cdot]$  into a symmetric (reversible) part  $\hat{\mathcal{J}}^S[\cdot]$  and anti-symmetric (irreversible) part  $\hat{\mathcal{J}}^A[\cdot]$  (denoting  $\hat{\mathcal{J}}^*[\cdot]$  as the adjoint operator of  $\hat{\mathcal{J}}[\cdot]$ ):

$$\begin{aligned}\hat{\mathcal{J}}^S[\varphi(\mathbf{z})] &= \frac{1}{2} \left( \hat{\mathcal{J}}[\varphi(\mathbf{z})] + \hat{\mathcal{J}}^*[\varphi(\mathbf{z})] \right) \\ &= \int_{\mathbb{R}^d} d\mathbf{x} \left[ \frac{1}{2} \left( W(\mathbf{z}|\mathbf{x})\pi(\mathbf{x}) + W(\mathbf{x}|\mathbf{z})\pi(\mathbf{z}) \right) \varphi(\mathbf{x}) \right] - \int_{\mathbb{R}^d} d\mathbf{x} [W(\mathbf{x}|\mathbf{z})\pi(\mathbf{z})\varphi(\mathbf{z})]; \\ \hat{\mathcal{J}}^A[\varphi(\mathbf{z})] &= \frac{1}{2} \left( \hat{\mathcal{J}}[\varphi(\mathbf{z})] - \hat{\mathcal{J}}^*[\varphi(\mathbf{z})] \right) = \int_{\mathbb{R}^d} d\mathbf{x} \left[ \frac{1}{2} \left( W(\mathbf{z}|\mathbf{x})\pi(\mathbf{x}) - W(\mathbf{x}|\mathbf{z})\pi(\mathbf{z}) \right) \varphi(\mathbf{x}) \right].\end{aligned}$$

By introducing a symmetric kernel function  $S(\mathbf{x}, \mathbf{z}) = S(\mathbf{z}, \mathbf{x}) = \frac{1}{2} \left( W(\mathbf{z}|\mathbf{x})\pi(\mathbf{x}) + W(\mathbf{x}|\mathbf{z})\pi(\mathbf{z}) \right)$ , and an anti-symmetric kernel  $A(\mathbf{x}, \mathbf{z}) = -A(\mathbf{z}, \mathbf{x}) = \frac{1}{2} \left( W(\mathbf{z}|\mathbf{x})\pi(\mathbf{x}) - W(\mathbf{x}|\mathbf{z})\pi(\mathbf{z}) \right)$ , we arrive at a different form of  $\mathcal{J}^S[\cdot]$  and  $\mathcal{J}^A[\cdot]$ :

$$\mathcal{J}^S[\varphi(\mathbf{z})] = \int_{\mathbb{R}^d} d\mathbf{x} [S(\mathbf{x}, \mathbf{z})\varphi(\mathbf{x}) - S(\mathbf{x}, \mathbf{z})\varphi(\mathbf{z})];$$

$$\mathcal{J}^A[\varphi(\mathbf{z})] = \int_{\mathbb{R}^d} d\mathbf{x} [A(\mathbf{x}, \mathbf{z})\varphi(\mathbf{x})].$$

Writing

$$\mathcal{J}[\varphi(\mathbf{z})] = \mathcal{J}^S[\varphi(\mathbf{z})] + \mathcal{J}^A[\varphi(\mathbf{z})] = \int_{\mathbb{R}^d} d\mathbf{x} [S(\mathbf{x}, \mathbf{z})\varphi(\mathbf{x}) - S(\mathbf{x}, \mathbf{z})\varphi(\mathbf{z}) + A(\mathbf{x}, \mathbf{z})\varphi(\mathbf{x})],$$

we can obtain the new formulation of the jump process as:

$$\frac{\partial p(\mathbf{z}|\mathbf{y}; t)}{\partial t} = \mathcal{J} \left[ \frac{p(\mathbf{z}|\mathbf{y}; t)}{\pi(\mathbf{z})} \right] = \int_{\mathbb{R}^d} d\mathbf{x} \left[ S(\mathbf{x}, \mathbf{z}) \frac{p(\mathbf{x}|\mathbf{y}; t)}{\pi(\mathbf{x})} - S(\mathbf{x}, \mathbf{z}) \frac{p(\mathbf{z}|\mathbf{y}; t)}{\pi(\mathbf{z})} + A(\mathbf{x}, \mathbf{z}) \frac{p(\mathbf{x}|\mathbf{y}; t)}{\pi(\mathbf{x})} \right].$$

Plugging  $p(\mathbf{z}|\mathbf{y}; t) = \pi(\mathbf{z})$  into the above equation, we find that as long as  $\int_{\mathbb{R}^d} A(\mathbf{x}, \mathbf{z}) d\mathbf{x} = 0$ ,  $\pi(\mathbf{z})$  is a stationary solution to the equation. Since  $\frac{S(\mathbf{x}, \mathbf{z}) + A(\mathbf{x}, \mathbf{z})}{\pi(\mathbf{x})}$  denotes a transition probability,  $S(\mathbf{x}, \mathbf{z}) + A(\mathbf{x}, \mathbf{z}) > 0$  for any  $\mathbf{x}$  and  $\mathbf{z}$ . We thereby notice that the requirement that  $\pi(\mathbf{z})$  is a stationary distribution of the jump process is translated into simpler constraints:  $\int_{\mathbb{R}^d} S(\mathbf{x}, \mathbf{z})\pi^{-1}(\mathbf{x})d\mathbf{x}$  and  $\int_{\mathbb{R}^d} A(\mathbf{x}, \mathbf{z})\pi^{-1}(\mathbf{x})d\mathbf{x}$  exists, with  $S(\mathbf{x}, \mathbf{z}) + A(\mathbf{x}, \mathbf{z}) > 0$ , and  $\int_{\mathbb{R}^d} A(\mathbf{x}, \mathbf{z})d\mathbf{x} = 0$ .

### C.2 Verifying condition 3 on $A(\mathbf{y}, \mathbf{y}^p, \mathbf{z}, \mathbf{z}^p)$ in Section 4.3

The anti-symmetric kernel  $A(\mathbf{y}, \mathbf{y}^p, \mathbf{z}, \mathbf{z}^p)$  (expressed in (28)) of (27) can be written as:

$$\begin{aligned}A(\mathbf{y}, \mathbf{y}^p, \mathbf{z}, \mathbf{z}^p) &= \frac{1}{2\Delta t} \left( \pi(\mathbf{y})\pi(\mathbf{y}^p)p(\mathbf{z}, \mathbf{z}^p|\mathbf{y}, \mathbf{y}^p; \Delta t) - \pi(\mathbf{z})\pi(\mathbf{z}^p)p(\mathbf{y}, \mathbf{y}^p|\mathbf{z}, \mathbf{z}^p; \Delta t) \right) \\ &= \frac{1}{2} \delta(\mathbf{z}^p - \mathbf{y}^p) \left( \mathfrak{F}(\mathbf{y}, \mathbf{y}^p, \mathbf{z}, \mathbf{z}^p) - \mathfrak{F}(\mathbf{z}, \mathbf{z}^p, \mathbf{y}, \mathbf{y}^p) \right) \\ &\quad - \frac{1}{2} \delta(\mathbf{z}^p + \mathbf{y}^p) \delta(\mathbf{z} - \mathbf{y}) \int_{\mathbb{R}^d} d\mathbf{x} \left( \mathfrak{F}(\mathbf{y}, \mathbf{y}^p, \mathbf{x}, -\mathbf{z}^p) - \mathfrak{F}(\mathbf{z}, \mathbf{z}^p, \mathbf{x}, -\mathbf{y}^p) \right) d\mathbf{x}.\end{aligned}$$

Below we prove that, as required,  $\int_{\mathbb{R}^{d+d^p}} A(\mathbf{y}, \mathbf{y}^p, \mathbf{z}, \mathbf{z}^p) d\mathbf{y} d\mathbf{y}^p = 0$ .

*Proof.*

$$\begin{aligned} & \int_{\mathbb{R}^{d+d^p}} A(\mathbf{x}, \mathbf{x}^p, \mathbf{z}, \mathbf{z}^p) d\mathbf{x} d\mathbf{x}^p \\ &= \frac{1}{2} \int_{\mathbb{R}^d} \left( \mathfrak{F}(\mathbf{y}, \mathbf{z}^p, \mathbf{z}, \mathbf{z}^p) - \mathfrak{F}(\mathbf{z}, \mathbf{z}^p, \mathbf{y}, \mathbf{z}^p) \right) d\mathbf{y} - \frac{1}{2} \int_{\mathbb{R}^d} \left( \mathfrak{F}(\mathbf{z}, -\mathbf{z}^p, \mathbf{x}, -\mathbf{z}^p) - \mathfrak{F}(\mathbf{z}, \mathbf{z}^p, \mathbf{x}, \mathbf{z}^p) \right) d\mathbf{x}. \end{aligned}$$

One can check that in (29) and (26),  $\tilde{f}(\mathbf{z}, \cdot | \mathbf{y}, -\mathbf{y}^p) = \tilde{g}(\mathbf{z}, \cdot | \mathbf{y}, \mathbf{y}^p)$ . Hence,  $\mathfrak{F}(\mathbf{y}, -\mathbf{y}^p, \mathbf{z}, -\mathbf{z}^p) = \mathfrak{F}(\mathbf{z}, \mathbf{z}^p, \mathbf{y}, \mathbf{y}^p)$ . Therefore

$$\begin{aligned} & \int_{\mathbb{R}^{d+d^p}} A(\mathbf{x}, \mathbf{x}^p, \mathbf{z}, \mathbf{z}^p) d\mathbf{x} d\mathbf{x}^p \\ &= \frac{1}{2} \int_{\mathbb{R}^d} \left( \mathfrak{F}(\mathbf{z}, -\mathbf{z}^p, \mathbf{y}, -\mathbf{z}^p) - \mathfrak{F}(\mathbf{z}, \mathbf{z}^p, \mathbf{y}, \mathbf{z}^p) \right) d\mathbf{y} - \frac{1}{2} \int_{\mathbb{R}^d} \left( \mathfrak{F}(\mathbf{z}, -\mathbf{z}^p, \mathbf{x}, -\mathbf{z}^p) - \mathfrak{F}(\mathbf{z}, \mathbf{z}^p, \mathbf{x}, \mathbf{z}^p) \right) d\mathbf{x} = 0. \end{aligned} \tag{43}$$

□

## D Proof of Theorem 2 (relation between forward process and adjoint process)

In this section, we use  $P(\mathbf{z}, T | \mathbf{y}, t)$  to denote the transition probability of the system being at state  $\mathbf{z}$  at time  $T$ , conditioning on it being at state  $\mathbf{y}$  at time  $t$  (i.e., for autonomous systems,  $P(\mathbf{z}, T | \mathbf{y}, t) = P(\mathbf{z} | \mathbf{y}; (T - t))$ ).

We first prove that for the infinitesimal generators, the backward probability transition kernel following the adjoint process and the forward probability transition kernel are related as:

$$\pi(\mathbf{y})P(\mathbf{z}, t + dt | \mathbf{y}, t) = \pi(\mathbf{z})P^\dagger(\mathbf{y}, t + dt | \mathbf{z}, t).$$

Taking path integrals with respect to the infinitesimal generators leads to the conclusion.

As is standard, we use two arbitrary smooth test functions  $\psi(\mathbf{y})$  and  $\phi(\mathbf{z})$ . Then

$$\begin{aligned} \int \int d\mathbf{y} d\mathbf{z} \psi(\mathbf{y}) \phi(\mathbf{z}) \frac{P(\mathbf{z}, t + dt | \mathbf{y}, t)}{\pi(\mathbf{z})} &= \int \int d\mathbf{y} d\mathbf{z} \psi(\mathbf{y}) \frac{\phi(\mathbf{z})}{\pi(\mathbf{z})} \left( P(\mathbf{z}, t | \mathbf{y}, t) + \frac{\partial P(\mathbf{z}, t | \mathbf{y}, t)}{\partial t} dt \right) \\ &= \int \int d\mathbf{y} d\mathbf{z} \psi(\mathbf{y}) \frac{\phi(\mathbf{z})}{\pi(\mathbf{z})} \left( P(\mathbf{z}, t | \mathbf{y}, t) + \mathcal{L}_{\mathbf{z}} \left[ \frac{P(\mathbf{z}, t | \mathbf{y}, t)}{\pi(\mathbf{z})} \right] dt \right), \end{aligned}$$

where  $\mathcal{L}_{\mathbf{z}}[\varphi(\mathbf{z})] = \nabla^T \cdot ([D(\mathbf{z}) + Q(\mathbf{z})] [\nabla \varphi(\mathbf{z}) \pi(\mathbf{z})])$  leads to the Fokker-Planck equation of the SDE. It can be checked using (13) and (14) that  $\mathcal{L}_{\mathbf{z}}^\dagger[\varphi(\mathbf{z})] = \mathcal{L}_{\mathbf{z}}^S[\varphi(\mathbf{z})] - \mathcal{L}_{\mathbf{z}}^A[\varphi(\mathbf{z})]$ .

For  $P^\dagger(\mathbf{y}, t + dt | \mathbf{z}, t)$ ,

$$\int \int d\mathbf{y} d\mathbf{z} \psi(\mathbf{y}) \phi(\mathbf{z}) \frac{P^\dagger(\mathbf{y}, t + dt | \mathbf{z}, t)}{\pi(\mathbf{y})} = \int \int d\mathbf{y} d\mathbf{z} \frac{\psi(\mathbf{y})}{\pi(\mathbf{y})} \phi(\mathbf{z}) \left( P^\dagger(\mathbf{y}, t | \mathbf{z}, t) + \mathcal{L}_{\mathbf{y}}^\dagger \left[ \frac{P^\dagger(\mathbf{y}, t | \mathbf{z}, t)}{\pi(\mathbf{y})} \right] dt \right).$$

Noting that  $P(\mathbf{z}, t | \mathbf{y}, t)$  and  $P^\dagger(\mathbf{y}, t | \mathbf{z}, t)$  equal to  $\delta(\mathbf{z} - \mathbf{y})$ , the zeroth order terms:

$$\int \int d\mathbf{y} d\mathbf{z} \psi(\mathbf{y}) \phi(\mathbf{z}) \frac{P(\mathbf{z}, t | \mathbf{y}, t)}{\pi(\mathbf{z})} = \int \int d\mathbf{y} d\mathbf{z} \psi(\mathbf{y}) \phi(\mathbf{z}) \frac{P^\dagger(\mathbf{y}, t | \mathbf{z}, t)}{\pi(\mathbf{y})}.$$

Then for the first order terms,

$$\begin{aligned}
& \int \int \mathrm{d}\mathbf{y} \mathrm{d}\mathbf{z} \psi(\mathbf{y}) \frac{\phi(\mathbf{z})}{\pi(\mathbf{z})} \mathcal{L}_{\mathbf{z}} \left[ \frac{P(\mathbf{z}, t | \mathbf{y}, t)}{\pi(\mathbf{z})} \right] \\
&= \int \int \mathrm{d}\mathbf{y} \mathrm{d}\mathbf{z} \psi(\mathbf{y}) \mathcal{L}_{\mathbf{z}}^{\dagger} \left[ \frac{\phi(\mathbf{z})}{\pi(\mathbf{z})} \right] \frac{P(\mathbf{z}, t | \mathbf{y}, t)}{\pi(\mathbf{z})} = \int \mathrm{d}\mathbf{z} \frac{\psi(\mathbf{z})}{\pi(\mathbf{z})} \mathcal{L}_{\mathbf{z}}^{\dagger} \left[ \frac{\phi(\mathbf{z})}{\pi(\mathbf{z})} \right] = \int \mathrm{d}\mathbf{z} \mathcal{L}_{\mathbf{z}} \left[ \frac{\psi(\mathbf{z})}{\pi(\mathbf{z})} \right] \frac{\phi(\mathbf{z})}{\pi(\mathbf{z})} \\
&= \int \int \mathrm{d}\mathbf{y} \mathrm{d}\mathbf{z} \mathcal{L}_{\mathbf{y}} \left[ \frac{\psi(\mathbf{y})}{\pi(\mathbf{y})} \right] \phi(\mathbf{z}) \frac{P^{\dagger}(\mathbf{y}, t | \mathbf{z}, t)}{\pi(\mathbf{y})} = \int \int \mathrm{d}\mathbf{y} \mathrm{d}\mathbf{z} \frac{\psi(\mathbf{y})}{\pi(\mathbf{y})} \phi(\mathbf{z}) \mathcal{L}_{\mathbf{y}}^{\dagger} \left[ \frac{P^{\dagger}(\mathbf{y}, t | \mathbf{z}, t)}{\pi(\mathbf{y})} \right].
\end{aligned}$$

Hence,

$$\begin{aligned}
\int \int \mathrm{d}\mathbf{y} \mathrm{d}\mathbf{z} \psi(\mathbf{y}) \phi(\mathbf{z}) \frac{P(\mathbf{z}, t + \mathrm{d}t | \mathbf{y}, t)}{\pi(\mathbf{z})} &= \int \int \mathrm{d}\mathbf{y} \mathrm{d}\mathbf{z} \psi(\mathbf{y}) \frac{\phi(\mathbf{z})}{\pi(\mathbf{z})} \left( P(\mathbf{z}, t | \mathbf{y}, t) + \mathcal{L}_{\mathbf{z}} \left[ \frac{P(\mathbf{z}, t | \mathbf{y}, t)}{\pi(\mathbf{z})} \right] \mathrm{d}t \right) \\
&= \int \int \mathrm{d}\mathbf{y} \mathrm{d}\mathbf{z} \frac{\psi(\mathbf{y})}{\pi(\mathbf{y})} \phi(\mathbf{z}) \left( P^{\dagger}(\mathbf{y}, t | \mathbf{z}, t) + \mathcal{L}_{\mathbf{y}}^{\dagger} \left[ \frac{P^{\dagger}(\mathbf{y}, t | \mathbf{z}, t)}{\pi(\mathbf{y})} \right] \mathrm{d}t \right) \\
&= \int \int \mathrm{d}\mathbf{y} \mathrm{d}\mathbf{z} \psi(\mathbf{y}) \phi(\mathbf{z}) \frac{P^{\dagger}(\mathbf{y}, t + \mathrm{d}t | \mathbf{z}, t)}{\pi(\mathbf{y})}.
\end{aligned}$$

Therefore, to the first order,

$$\pi(\mathbf{y}) P(\mathbf{z}, t + \mathrm{d}t | \mathbf{y}, t) = \pi(\mathbf{z}) P^{\dagger}(\mathbf{y}, t + \mathrm{d}t | \mathbf{z}, t).$$

Using the Markov properties,

$$P(\mathbf{z}_N, t_N | \mathbf{z}_0, t_0) = \int \cdots \int \prod_{i=1}^{N-1} \mathrm{d}\mathbf{z}_i \prod_{i=0}^{N-1} P(\mathbf{z}_{i+1}, t_{i+1} | \mathbf{z}_i, t_i);$$

and

$$\begin{aligned}
P^{\dagger}(\mathbf{z}_0, t_N | \mathbf{z}_N, t_0) &= \int \cdots \int \prod_{i=1}^{N-1} \mathrm{d}\mathbf{z}_i \prod_{i=0}^{N-1} P^{\dagger}(\mathbf{z}_i, t_{i+1} | \mathbf{z}_{i+1}, t_i) \\
&= \int \cdots \int \prod_{i=1}^{N-1} \mathrm{d}\mathbf{z}_i \prod_{i=0}^{N-1} \frac{\pi(\mathbf{z}_i)}{\pi(\mathbf{z}_{i+1})} P(\mathbf{z}_{i+1}, t_{i+1} | \mathbf{z}_i, t_i) \\
&= \int \cdots \int \prod_{i=1}^{N-1} \mathrm{d}\mathbf{z}_i \prod_{i=0}^{N-1} \frac{\pi(\mathbf{z}_i)}{\pi(\mathbf{z}_{i+1})} P(\mathbf{z}_{i+1}, t_{i+1} | \mathbf{z}_i, t_i) = \frac{\pi(\mathbf{z}_0)}{\pi(\mathbf{z}_N)} P(\mathbf{z}_N, t_N | \mathbf{z}_0, t_0).
\end{aligned}$$

Taking the time interval between  $t_i$  and  $t_{i+1}$  to be infinitesimal, we obtain that  $\frac{P(\mathbf{z}^{(T)}, T | \mathbf{z}^{(t)}, t)}{P^{\dagger}(\mathbf{z}^{(t)}, T | \mathbf{z}^{(T)}, t)} = \frac{\pi(\mathbf{z}^{(T)})}{\pi(\mathbf{z}^{(t)})}$ .

Analysis on the semigroups  $e^{t\mathcal{L}}$  and  $e^{t\mathcal{L}^{\dagger}}$  generated by  $\mathcal{L}$  and  $\mathcal{L}^{\dagger}$  can also lead to this conclusion.

## E Experiments

### E.1 Parameter settings for the irreversible jump sampler

In the 1D experiments, we use  $\beta = 1.2$  for the normal distribution case (where the length of the region of definition is 10) and  $\beta = 0.8$  for the log-normal distribution (where the length of the region of definition is 5). The acceptance

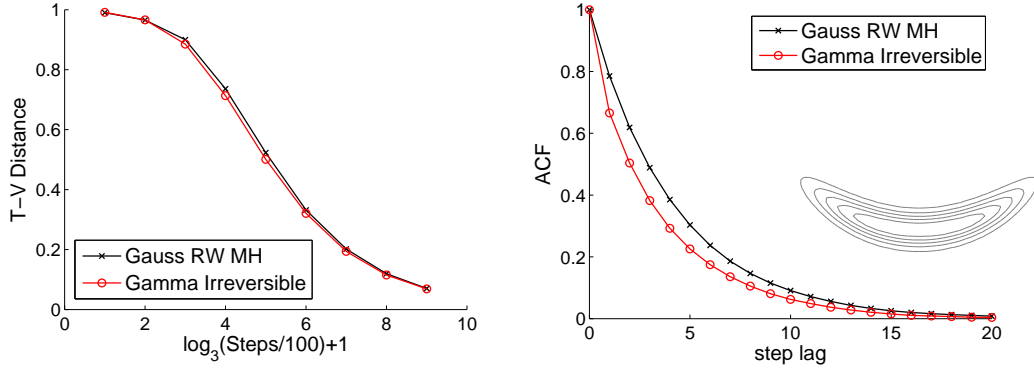


Figure 8: Moon-shaped distribution. T-V distance and ACF of Gauss RW MH and the new irreversible Gamma sampler in terms of number of steps.

rate is around 50% in these cases. Due to the irreversibility of the sampler, a high acceptance rate can be maintained while reducing the autocorrelation time.

In the visual comparison of samplers of Fig. 2, we use  $\beta = 0.15$  (where the lengths of the region of definition is  $2 \times 2$ ). In the 2D bimodal experiments, we use  $\beta = 0.4$  (where the lengths of the region of definition is  $6 \times 3$ ). In the 2D multimodal experiments, we take  $\beta = 1.5$  (where the lengths of the region of definition is  $14 \times 14$ ). For the 2D correlated distribution, we take  $\beta = 0.25$  (where the lengths of the region of definition is  $5 \times 2$ ).

## E.2 Further results for the correlated distribution case

For the correlated distribution case, we compare the irreversible sampler of Algorithm 4 with the Gaussian random walk MH method. We find that in terms of number of steps, the irreversible jump sampler decorrelates faster and converges in fewer steps. See Fig. 8, which provides the same comparison as in Fig. 7, but here in terms of number of iterations rather than runtime.

## References

- [Bardenet et al., 2014] Bardenet, R., Doucet, A., and Holmes, C. (2014). Towards scaling up Markov chain Monte Carlo: An adaptive subsampling approach. In *Proceedings of the 30th International Conference on Machine Learning (ICML’14)*.
- [Bardenet et al., 2015] Bardenet, R., Doucet, A., and Holmes, C. (2015). On Markov chain Monte Carlo methods for tall data.
- [Bierkens, 2015] Bierkens, J. (2015). Non-reversible Metropolis-Hastings. *J. Stat. Comput.*, pages 1–16.
- [Bierkens et al., 2016] Bierkens, J., Fearnhead, P., and Roberts, G. (2016). The Zig-Zag process and super-efficient sampling for Bayesian analysis of big data.
- [Bierkens and Roberts, 2016] Bierkens, J. and Roberts, G. (2016). A piecewise deterministic scaling limit of Lifted Metropolis-Hastings in the Curie-Weiss model.
- [Bou-Rabee and Owhadi, 2010] Bou-Rabee, N. and Owhadi, H. (2010). Long-run accuracy of variational integrators in the stochastic context. *SIAM J. Num. Anal.*, 48:278–297.
- [Bouchard-Côté et al., 2016] Bouchard-Côté, A., Vollmer, S. J., and Doucet, A. (2016). The bouncy particle sampler: A non-reversible rejection-free Markov chain Monte Carlo method.
- [Chen et al., 2015] Chen, C., Ding, N., and Carin, L. (2015). On the convergence of stochastic gradient mcmc algorithms with high-order integrators. In *Advances in Neural Information Processing Systems 28*, pages 2278–2286.
- [Chen et al., 1999] Chen, F., Lovász, L., and Pak, I. (1999). Lifting Markov chains to speed up mixing. In *Proceedings of the 31st annual ACM STOC*, pages 275–281.
- [Chen et al., 2014] Chen, T., Fox, E. B., and Guestrin, C. (2014). Stochastic gradient Hamiltonian Monte Carlo. In *Proceeding of 31st International Conference on Machine Learning (ICML’14)*.
- [Chen and Hwang, 2013] Chen, T.-L. and Hwang, C.-R. (2013). Accelerating reversible Markov chains. *Statistics & Probability Letters*, 83(9):1956–1962.
- [Chib and Greenberg, 1995] Chib, S. and Greenberg, E. (1995). Understanding the Metropolis-Hastings algorithm. *The American Statistician*, 49(4):327–335.
- [Diaconis et al., 2000] Diaconis, P., Holmes, S., and Neal, R. M. (2000). Analysis of a nonreversible Markov chain sampler. *Ann. Appl. Probab.*, 10:726–752.
- [Ding et al., 2014] Ding, N., Fang, Y., Babbush, R., Chen, C., Skeel, R. D., and Neven, H. (2014). Bayesian sampling using stochastic gradient thermostats. In *Advances in Neural Information Processing Systems 27 (NIPS’14)*.
- [Duane et al., 1987] Duane, S., Kennedy, A. D., Pendleton, B. J., and Roweth, D. (1987). Hybrid Monte Carlo. *Physics Letters B*, 195(2):216 – 222.
- [Duncan et al., 2016] Duncan, A. B., Lelièvre, T., and Pavliotis, G. A. (2016). Variance reduction using nonreversible Langevin samplers. *Journal of Statistical Physics*, 163(3):457–491.
- [Durmus et al., 2016] Durmus, A., Roberts, G. O., Vilmart, G., and Zygalakis, K. C. (2016). Fast Langevin based algorithm for MCMC in high dimensions.

- [Fang et al., 2014] Fang, Y., Sanz-Serna, J. M., , and Skeel, R. D. (2014). Compressible generalized hybrid Monte Carlo. *J. Chem. Phys.*, 140:174108.
- [Gardiner, 2009] Gardiner, C. W. (2009). *Handbook of Stochastic Methods*. Springer, 4th edition.
- [Girolami and Calderhead, 2011] Girolami, M. and Calderhead, B. (2011). Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(2):123–214.
- [Gustafson, 1998] Gustafson, P. (1998). A guided walk Metropolis algorithm. *Statistics and Computing*, 8(4):357–364.
- [Hastings, 1970] Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57:pp 97–109.
- [Horowitz, 1991] Horowitz, A. M. (1991). A generalized guided Monte Carlo algorithm. *Physics Letters B*, 268(2):247 – 252.
- [Hwang et al., 1993] Hwang, C.-R., Hwang-Ma, S.-Y., and Sheu, S.-J. (1993). Accelerating Gaussian diffusions. *Ann. Appl. Probab.*, 3(3):897–913.
- [Hwang et al., 2005] Hwang, C.-R., Hwang-Ma, S.-Y., and Sheu, S.-J. (2005). Accelerating diffusions. *Ann. Appl. Probab.*, 15(2):1433–1444.
- [Jarner and Roberts, 2007] Jarner, S. F. and Roberts, G. O. (2007). Convergence of heavy-tailed Monte Carlo Markov chain algorithms. *Scandinavian Journal of Statistics*, 34(4):781–815.
- [Korattikara et al., 2014] Korattikara, A., Chen, Y., and Welling, M. (2014). Austerity in MCMC land: Cutting the Metropolis-Hastings budget. In *Proceedings of the 30th International Conference on Machine Learning (ICML’14)*.
- [Kou et al., 2006] Kou, S. C., Zhou, Q., and Wong, W. H. (2006). Discussion paper: Equi-energy sampler with applications in statistical inference and statistical mechanics. *The Annals of Statistics*, 34(4):1581–1619.
- [Leimkuhler et al., 2014] Leimkuhler, B., Matthews, C., and Tretyakov, M. (2014). On the long-time integration of stochastic gradient systems. *Proceedings of the Royal Society A*, 470:20140120.
- [Liu, 2004] Liu, J. S. (2004). *Monte Carlo Strategies in Scientific Computing*. Springer-Verlag.
- [Ma et al., 2015] Ma, Y.-A., Chen, T., and Fox, E. (2015). A complete recipe for stochastic gradient MCMC. In *Advances in Neural Information Processing Systems* 28, pages 2899–2907.
- [Ma and Qian, 2015] Ma, Y.-A. and Qian, H. (2015). Universal ideal behavior and macroscopic work relation of linear irreversible stochastic thermodynamics. *New Journal of Physics*, 17(6):065013.
- [Metropolis et al., 1953] Metropolis, M., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and E., T. (1953). Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21.
- [Neal, 2004] Neal, R. M. (2004). Improving asymptotic variance of MCMC estimators: Non-reversible chains are better.
- [Neal, 2010] Neal, R. M. (2010). MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 54:113–162.

- [Ottobre et al., 2016] Ottobre, M., Pillai, N. S., Pinski, F. J., and Stuart, A. M. (2016). A function space HMC algorithm with second order Langevin diffusion limit. *Bernoulli*, 22(1):60–106.
- [Patterson and Teh, 2013] Patterson, S. and Teh, Y. W. (2013). Stochastic gradient Riemannian Langevin dynamics on the probability simplex. In *Advances in Neural Information Processing Systems 26 (NIPS’13)*.
- [Rey-Bellet and Spiliopoulos, 2015] Rey-Bellet, L. and Spiliopoulos, K. (2015). Irreversible Langevin samplers and variance reduction: A large deviations approach. *Nonlinearity*, 28(7):2081.
- [Robbins and Monro, 1951] Robbins, H. and Monro, S. (1951). A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407.
- [Robert and Casella, 2004] Robert, C. and Casella, G. (2004). *Monte Carlo statistical methods*. Springer Science & Business Media, 2nd edition.
- [Roberts and Stramer, 2002] Roberts, G. O. and Stramer, O. (2002). Langevin diffusions and Metropolis-Hastings algorithms. *Methodology And Computing In Applied Probability*, 4:337–357.
- [Shang et al., 2015] Shang, X., Zhu, Z., Leimkuhler, B., and Storkey, A. (2015). Covariance-controlled adaptive Langevin thermostat for large-scale Bayesian sampling. In *Advances in Neural Information Processing Systems 28 (NIPS’15)*.
- [Shi et al., 2012] Shi, J., Chen, T., Yuan, R., Yuan, B., and Ao, P. (2012). Relation of a new interpretation of stochastic differential equations to Itô process. *Journal of Statistical Physics*, 148(3):579–590.
- [Tak et al., 2016] Tak, H., Meng, X.-L., and van Dyk, D. A. (2016). A repulsive-attractive Metropolis algorithm for multimodality.
- [Tuckerman et al., 2001] Tuckerman, M., Liu, Y., Ciccotti, G., and Martyna, G. (2001). Non-Hamiltonian molecular dynamics: Generalizing Hamiltonian phase space principles to non-Hamiltonian systems. *J. Chem. Phys.*, 115:1678–1702.
- [Turitsyn et al., 2011] Turitsyn, K. S., Chertkov, M., and Vucelja, M. (2011). Irreversible Monte Carlo algorithms for efficient sampling. *Physica D: Nonlinear Phenomena*, 240(4–5):410–414.
- [Vucelja, 2015] Vucelja, M. (2015). Lifting – a nonreversible Markov chain Monte Carlo algorithm.
- [Welling and Teh, 2011] Welling, M. and Teh, Y. W. (2011). Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning (ICML’11)*, pages 681–688.
- [Xifara et al., 2014] Xifara, T., Sherlock, C., Livingstone, S., Byrne, S., and Girolami, M. (2014). Langevin diffusions and the Metropolis-adjusted Langevin algorithm. *Statistics and Probability Letters*, 91:14–19.



